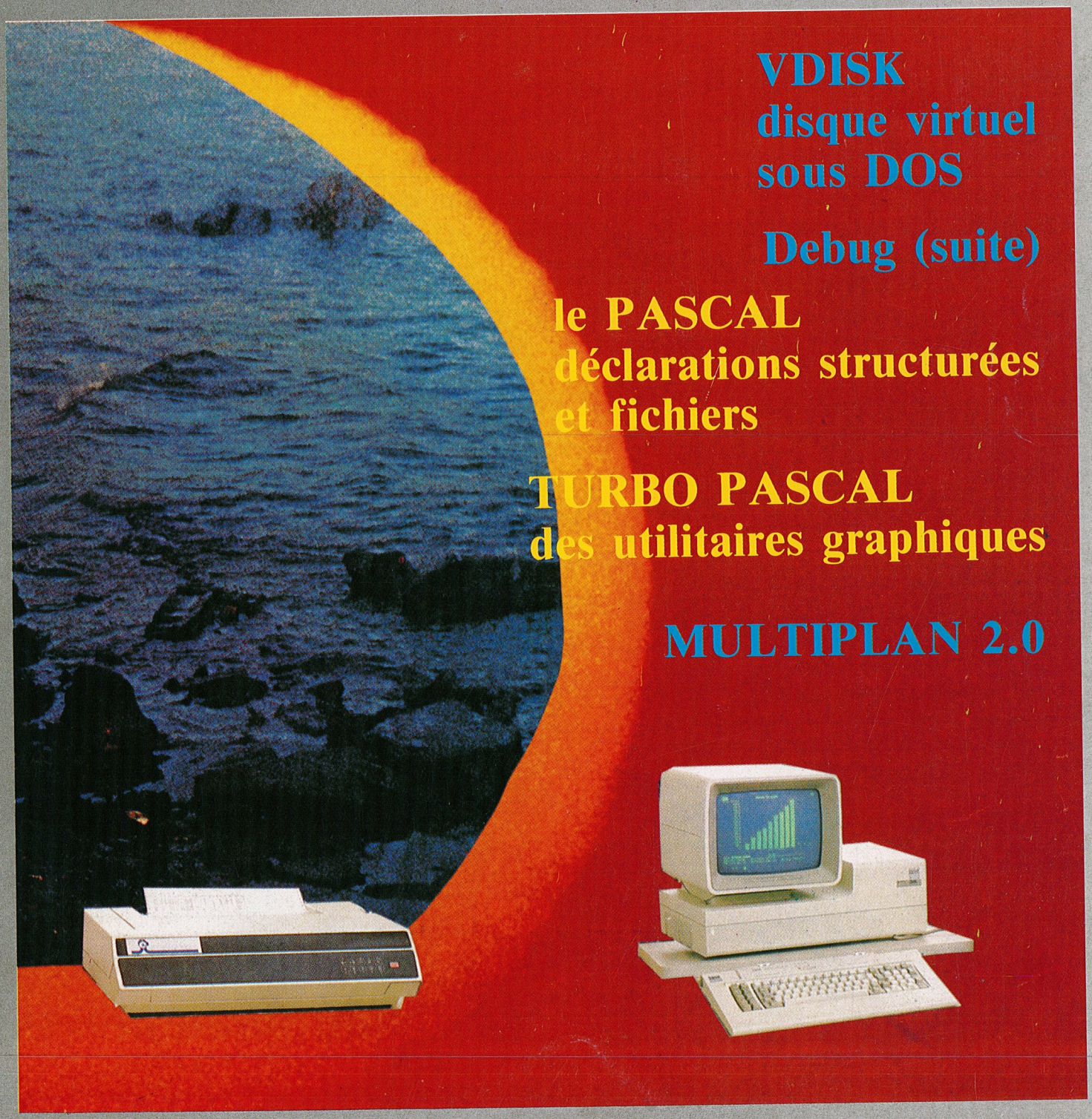
ISSN 0706-0775 X

M 2537 - N° 25 - 21 F

La Revue de l'Utilisateur de l'IBM-PC

Le magazine de l'IBM-PC et de ses compatibles

Juin-Juillet 1986 - N° 25 - 21 F
Suisse 6 FS - Belgique 160 FB - Canada 3,95 $ Can.



VDISK
disque virtuel sous DOS

Debug (suite)

le PASCAL
déclarations structurées et fichiers

TURBO PASCAL
des utilitaires graphiques

MULTIPLAN 2.0

PÉRI TECHNOLOGIE DONNE LE TEMPO

Grâce à une innovation technologique permanente pour toujours plus de capacité, toujours plus de compatibilité et de performances, PÉRI TECHNOLOGIE donne un rythme à l'univers P.C. : FORTISSIMO

PÉRI TECHNOLOGIE, LA MAESTRIA INFORMATIQUE.

PÉRI technologie

PÉRI TECHNOLOGIE
15, ALLÉE DES PLATANES, SOFILIC 437
94263 FRESNES CEDEX - TÉL. (1) 46.66.06.31 - TÉLEX : 270 239 F

AGENCE RHÔNE-ALPES
IMMEUBLE "LE PRIMAT" - 190 RUE GARIBALDI,
69003 LYON - TÉL. 78.62.75.05 - TÉLEX : 306 092 F

EDITO

Une profession de foi

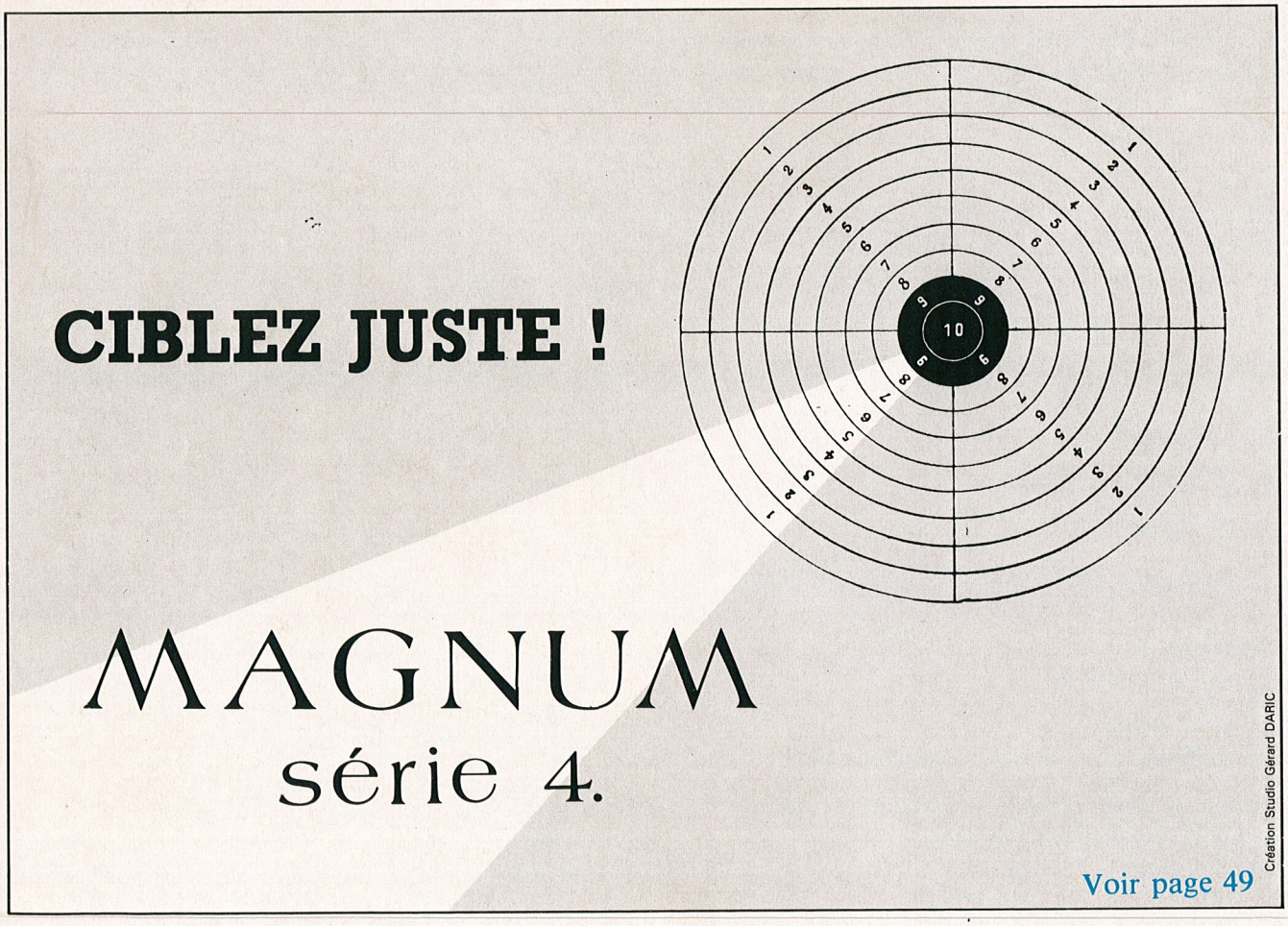
Roger CHRISTOPHE

Ce numéro 25, daté Juin-Juillet 1986 comporte la fin de la série Debug, notamment en ce qui concerne le contrôle de périphériques ainsi que les fonctions supplémentaires que comporte le DOS. Dans notre prochain numéro daté de Septembre 1986, J.P. Pruniaux commencera une nouvelle série concernant les Adresses Utiles et son premier article sera consacré à l'apprentissage des commandes du haut-parleur.
Notre série sur Turbo Pascal continuera également en Septembre, et, nous vous annonçons qu'à partir de la rentrée, la Revue de l'Utilisateur de l'IBM-PC sera en kiosque en début de mois.

Septembre 86 pourrait être, selon les « prophéties » des observateurs de la micro-informatique professionnelle, une date charnière pour son développement. Il semble que nous allions vers une « banalisation » du micro-ordinateur et les baisses de prix observées confortent cette tendance. Aussi la nouvelle mode est aux logiciels destinés aux non-informaticiens, et nous avons vu apparaître certains d'entre-eux qui ne requièrent aucune connaissance particulière en informatique.

Mais, à notre sens, nous ne pensons pas qu'ils puissent détrôner les logiciels plus techniques, car la programmation est un jeu et l'activité ludique est le propre de l'homme.
De plus, le fait de savoir programmer, donne une plus grande liberté de création de ses propres applications et permet, de ce fait, de faire reculer les limites de son PC.

Et cela reste notre profession de foi et la ligne d'action de notre politique rédactionnelle.


CIBLEZ JUSTE !
MAGNUM
série 4.
Création Studio Gérard DARIC
Voir page 49

SOMMAIRE

4 VDISK

VDISK ou comment créer un disque virtuel sous la version 3 du DOS. Daniel ARCANES traite des trois principales catégories de cartes d'extension.

17 DEBUG

DEBUG, système et assembleur. Contrôle d'Entrée Sortie. Jean-Paul PRUNIAUX, après avoir passé en revue la plupart des fonctions du DOS 2.0 développe maintenant les fonctions périphériques et les fonctions supplémentaires du DOS 3.0.

27 LE PASCAL

LE PASCAL, déclarations structurées et fichiers Jean-Paul PRUNIAUX poursuit son analyse du PASCAL, avec quelques déclarations structurées supplémentaires et des fichiers.

42 TURBO

Sauvegarde et Impression d'Ecrans Graphiques sur IBM-PC, en TURBO Jean-Pierre CHARLES présente deux solutions destinées à fixer des images sur papier ou sur disquette pour les amateurs de TURBO-PASCAL.

49 MULTIPLAN

Multiplan 2.0. Vous êtes français et vous possédez un PC, Daniel ARCANES donne des informations sur les macros.

La Revue de l'Utilisateur de l'IBM-PC

Le magazine de l'IBM-PC et de ses compatibles

Mensuel 21 F
2, rue d'Amsterdam
75009 Paris
Tél. : (1) 42.81.54.27.
Directrice de la publication : Sardie Errera
Rédacteur en chef : Roger Christophe
édité par SOFTIN COMMUNICATION (France)
Sarl R.C. Paris B 329 834 592
N° Siret 329 834 592 000 19
Commission Paritaire (en cours)
Dépôt légal : 2e trimestre 1986

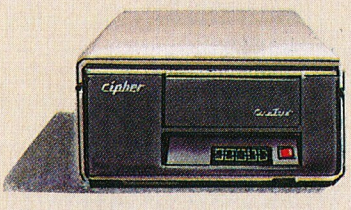
CCP Paris 8850 84 D
Composition : SOFTIN COMMUNICATION
Impression : MS (Montreuil)
Les marques IBM et IBM-PC sont des marques déposées de la compagnie IBM
La société éditrice SOFTIN COMMUNICATION (France) déclare que la Revue de l'Utilisateur de l'IBM-PC n'a aucun lien avec la Compagnie IBM et que sont indépendance rédactionnelle est totale.
Accord spécial avec Publi T Service (PTS)

Streamer F 9000

PC/GROS SYSTÈMES: LA COHABITATION via les bandes 1/2 pouce

Le transfert de données entre micro-ordinateur IBM ou compatibles et Mainframe est désormais possible grâce au streamer CIPHER F 9000.
En effet, une bande 1/2 pouce 1600 ou 6250 bpi écrite en EBCDIC comme en ASCII pourra être relue par le F 9000 et transférée sur le disque dur d'un micro-ordinateur et inversement.

SFCE
PERI INFORMATIQUE
8, Avenue Léon Harmel - 92160 ANTONY
Tél. 46 66 21 62 - Télex: 201.139 F



cipher data products, sarl
2/4, Avenue de la Cerisaie, Platanes 305
94266 FRESNES CEDEX - Tél. 46 68 87 87
Télex: CIPHERF 203 935

SAGHA

COUPON-RÉPONSE A RENVOYER A SFCE

NOM _____ FONCTION _____ SOCIÉTÉ _____

ADRESSE _____ TÉL. _____

souhaite recevoir des informations complémentaires sur le F 9000

RU

VDISK ou comment créer un disque virtuel sous la version 3 du DOS

Les cartes d'extension mémoire apparues depuis un certain temps se divisent en trois catégories principales qui se recoupent plus ou moins.

Daniel ARCANES

Leur véritable point commun est cependant assez clair, la Vitesse.
Quelque soit le mode d'utilisation, les cartes d'extension *RAM* amènent le mode d'utilisation, les cartes d'extension *RAM* amènent le *PC* à travailler beaucoup plus vite, c'est déjà une raison majeure pour l'utilisateur d'un *PC*.
En effet, dans tous les cas de figure, les cartes d'extension mémoire, per-

NOUVEAU

RENDEZ VOS STATISTIQUES MAGIQUES!

avec SPSS/PC™

Logiciel de statistiques et d'édition de rapport pour IBM / PC et compatibles

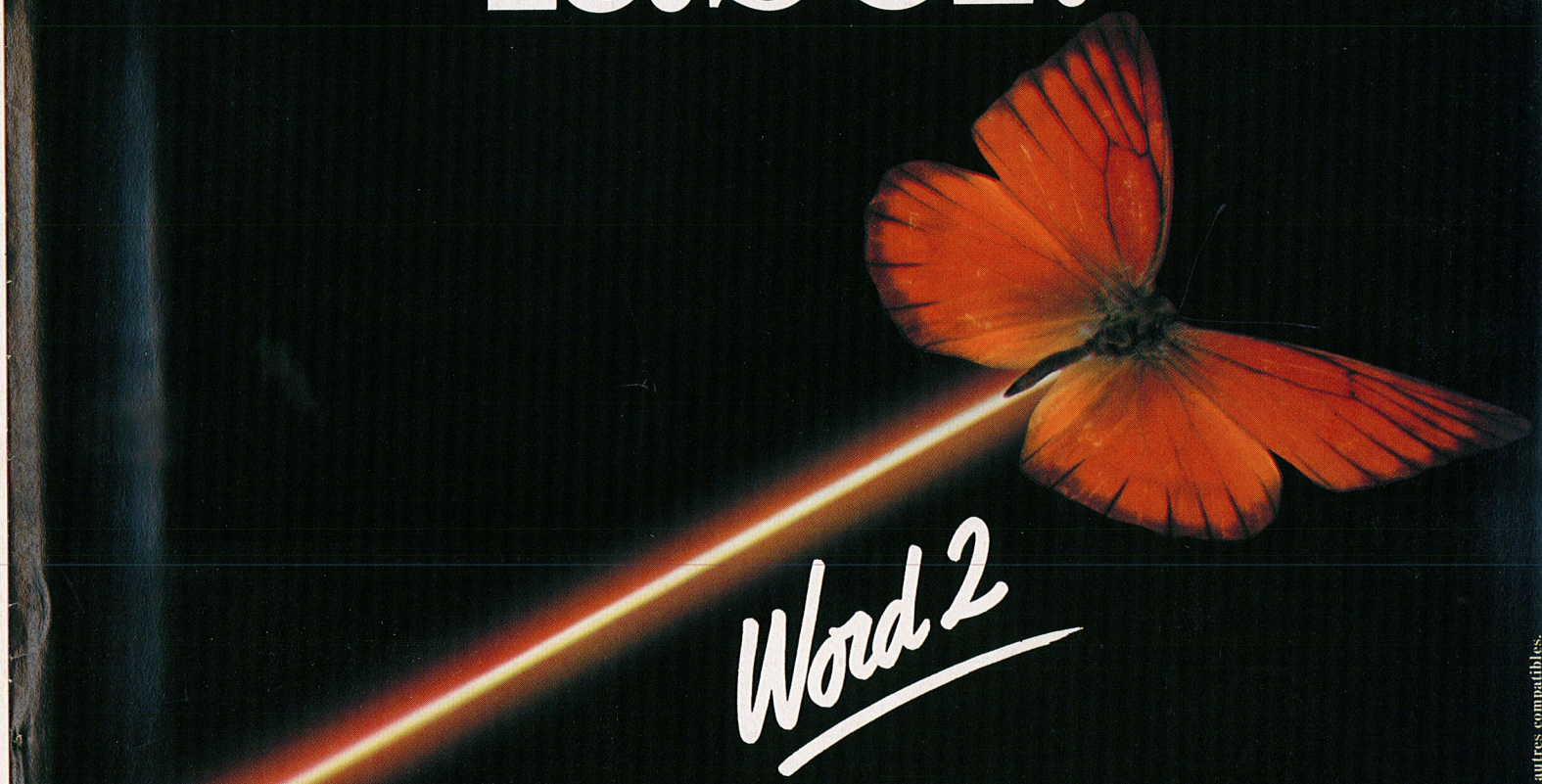
- Tableaux croisés.
- Analyse de variance.
- Régression multiple.
- Plus de 25 procédures intégrées.
- Gestion de fichiers.
- Interconnexion avec d'autres logiciels.
- Affichage automatique ou sur mesure.
- Tableaux entièrement référencés.
- Tracés et graphiques.

Livré avec une documentation en français pour tous niveaux d'utilisateurs.

Renseignements

37, rue des Acacias, 75017 Paris.
Tél. (1) 47.64.14.81

L'écriture laser.



Word écrivait comme l'éclair. Pour Microsoft, ce n'était pas assez. Place à Microsoft Word 2, le traitement de texte laser.

Aussi simple d'emploi que Word version 1, il est encore plus puissant et plus riche fonctionnellement. Et surtout, il est spécialement conçu pour pousser à l'extrême les performances des nouvelles imprimantes laser, matricielles, jet d'encre et marguerite. Tous les records de vitesse, de silence et de qualité d'impression sont dépassés.

Microsoft Word 2, un progrès fulgurant pour les IBM PC et compatibles.*

MICROSOFT®

Les logiciels de la vie simple.

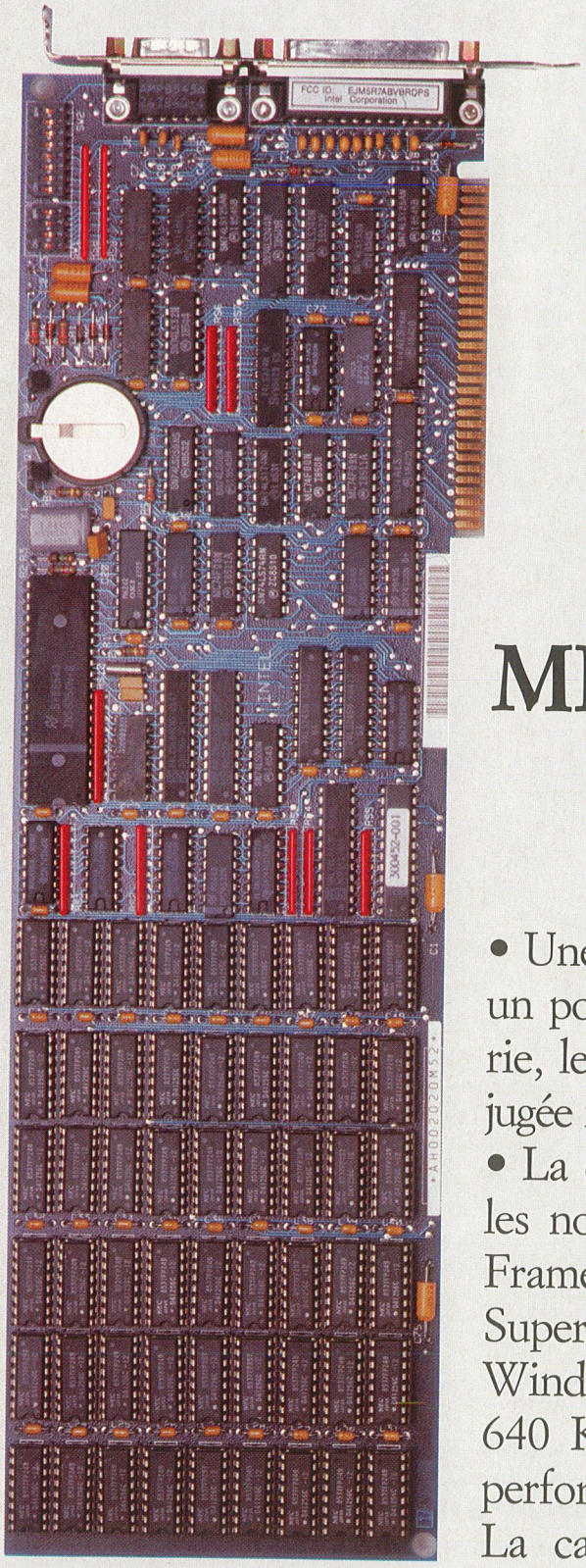
N° 519 Local Québec 91946 - Les Ulis Cedex.

*Bull Micral, Commodore, Goupil, HP Vectra, Léanord, Olivetti et autres compatibles.

RSCG Technologies

CARTE INTEL 1,5 Mo

Pour PC et XT



MULTIFONCTION ET MEMOIRE ETENDUE

• Une carte multifonction combinant : un port série, un port parallèle, une horloge sauvegardée par batterie, les logiciels pour imprimante et RAM disque, est jugée indispensable par de nombreux utilisateurs de PC.

• La carte mémoire INTEL étendue à 1,5 Mo pour les nouveaux logiciels :
Framework II, Lotus 1-2-3 (Version 2), Open-Access, Supercalc 3 (Version 2.1), Symphony (Version 1.1), Windows, est indispensable pour dépasser la limite de 640 K du DOS et augmenter considérablement les performances des logiciels.

La carte Intel multifonction 1,5 Mo vous donne toutes ces fonctions sur une seule carte, pour tirer le meilleur parti du PC.

7.750 F (HT)

FRAMEWORK, LOTUS 1-2-3 OPEN-ACCESS, SUPERCALC, SYMPHONY ET WINDOWS sont des marques déposées.

La Commande Electronique

intel

mettront, outre le fait d'amener la mémoire du *PC* à 640 K, si ce n'était pas le cas, et donc d'accepter tous les logiciels écrits pour le *PC*, ainsi que de pouvoir disposer d'un disque virtuel sur le *PC*.

DISQUE VIRTUEL

Un disque virtuel va simuler un véritable disque.
En utilisant *VDISK* avec le VDISK DOS 3.0. nous pourrons indiquer au système qu'une certaine partie de la mémoire sera désormais considérée comme l'unité B:, C: ou D: du *PC*.
En fait, le *DOS* réserve une partie de la *RAM* à ce disque virtuel et y accède comme s'il s'agissait d'un disque.
Nous pourrons alors recopier des fichiers dans ce disque virtuel, parfois, également appelé *RAM* disque.
Avant d'appeler un logiciel, avec pour conséquence, des gains de vitesse de traitement assez importants, la gestion de fichier s'effectuant en mémoire centrale et donc, en général, beaucoup plus rapidement que lorsque des entrées-sorties disque sont impliquées.
Alors pourquoi cette différence de vitesse ? Il y a deux explications à cela.
D'abord, vous le savez sans doute, en général, la *RAM* de base disponible (disons, par exemple 100 K) est souvent insuffisante pour charger l'intégralité d'un fichier, ainsi par conséquent, lorsque l'ordinateur va effectuer certaines opérations, tel qu'un tri, il va charger des parties du fichier dans la *RAM*, les recopier sur disque, en charger d'autres dans la *RAM* et les traiter. Ce va et vient entre la *RAM* et le disque peut se révéler de ce fait assez long.

Si, en revanche, le disque en question est lui-même présent dans la *RAM*, un certain temps sera dès lors gagné car les échanges se feront entre une portion de la *RAM*, celle qui contient notamment le fichier et une autre, celle gérée par le programme.

Or, lorsque le microprocesseur travaille avec la *RAM*, il dépose et décode dans celle-ci des charges électriques de base (1 et 0). Cette opération s'effectue presque instantanément au niveau électronique. Cependant, ces charges de 0 et 1 qui servent à représenter les informations disparaissent lorsque le *PC* est éteint, ce qui explique la nécessité des sauvegardes sur un support durable et magnétique, les disquettes par exemple.

Les imprimantes de l'écriture laser.

Imprimantes supportées par Microsoft Word 2.

Imprimantes Laser
• Apple LaserWriter • Canon A1/A2 Laser • HP LaserJet/LaserJet +.

Imprimantes Matricielles
• Brother HRX5 • Canon PW 1080 A •C-Itoh 3500 • Epson FX 85, 105 • Epson LQ 1500 et RX 100 • Facit 4511, 4512, 4513, 4514 • Honeywell LQ 34 • Ibmgraphics printer • Ibm 3852, 4201 • Mannesmann Tally 85, 86, 180, 280 • Nec P2, P3 • Oki 192, 193 • Olivetti PR 15, 320, 340, 17 B • Star • Texas Instruments 855 • Toshiba P 1350, 1351.

Imprimantes Jet d'encre
• Canon A 40 • Siemens PT 88, 89.

Imprimantes Marguerite
• Daisywriter 2000 • Diablo 630 ECS • Facit 4565 • Ibm 5216, 6747 • Nec Serie Spinwriter • Olivetti PR 320 B, 340 B, Serie ET • Olympia ESW 3000K, ESW 1000 • Qume Letterpro 20 et Serie Sprint.

Microsoft Word 2 supporte également toutes les imprimantes compatibles avec celles citées ci-dessus.

Pour recevoir une documentation sur Microsoft Word 2, renvoyez ce coupon-réponse à Microsoft - Local Québec N° 519 - 91946 Les Ulis Cedex.

Nom : _____ Prénom : _____
Fonction : _____
Société : _____
Adresse : _____
_____ Tél. : _____

MICROSOFT®
Les logiciels de la vie simple.

Créez votre solution

Turbo Pascal et ses Toolbox

Pour chaque domaine d'application, Borland a conçu une bibliothèque d'outils de programmation prêts à l'emploi. Vous êtes sûr d'y trouver de quoi créer votre solution – celle qui correspond exactement à vos envies et vos besoins spécifiques.

Plus qu'un langage

Turbo Pascal est un environnement complet de programmation extrêmement rapide grâce à la compilation en mémoire, et facile d'emploi. Les débutants comme les professionnels peuvent maintenant programmer en Pascal, l'un des langages les plus performants sur micro-ordinateurs. Plus de 500.000 utilisateurs dans le monde en sont enthousiastes! Turbo Pascal et ses utilitaires sont disponibles (en français) sur la plupart des micros, depuis l'Amstrad jusqu'aux AT – et nous sortons constamment de nouvelles adaptations. Appelez-nous pour connaître celles qui tournent sur votre machine!

Des applications multiples

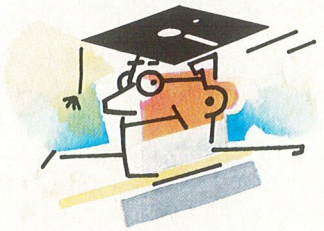
Gestion, calculs, enseignement, graphisme, IA, jeux – on peut tout faire en Turbo Pascal. Il a été choisi dans le plan "Informatique pour Tous" pour l'enseignement du Pascal; une Université s'en est servi pour développer un important logiciel de DAO (dessin assisté par ordinateur). Tecsi Software a créé un système expert écrit en Turbo Pascal, Asforgid un intégré comprenant traitement de texte, base de données et graphisme – pour n'en citer que quelques exemples...

Les Toolbox

Ce sont des utilitaires de Turbo Pascal et couvrent chacun un domaine d'application précis. Un Toolbox comprend une collection de routines en code source sur la disquette, un manuel et un exemple prêt à être compilé. Vous l'utilisez comme tel, ou alors vous profitez des routines déjà écrites pour créer vos propres programmes sur mesure. (Vos programmes compilés sont librement commercialisables).

Turbo Pascal

Turbo Pascal intègre un éditeur plein écran, un compilateur et un debuggeur en un seul programme. Sa rapidité vient de la compilation en mémoire. De plus, le compilateur retrouve automatiquement l'emplacement d'une erreur dans le code source, ce qui facilite et accélère la mise au point de vos programmes.
Les machines 16 bits disposent de deux options:
l'option 87 pour gérer le 8087,
l'option BCD (décimal code binaire) pour les calculs de gestion.
Les versions IBM et Amstrad option graphique comprennent des routines graphiques et une tortue.
Turbo Pascal est livré avec le code source de MicroCalc, petit tableur prêt à l'emploi.



Turbo Tutor

Cette méthode d'auto-formation accessible à tout le monde vous initiera aux bons usages de la programmation en Turbo Pascal, progressivement et tout en douceur. Ça se corse un peu au fur et à mesure, et même les programmeurs confirmés y trouveront quelques astuces bien pratiques pour leurs propres développements. (Un manuel + une disquette d'exemples).

Turbo Editor

Pour construire le traitement de texte dont vous avez toujours rêvé, ou pour inclure un éditeur dans vos programmes – toutes les routines sont dans Editor Toolbox. Vous pouvez notamment :

- changer les commandes de l'éditeur,
- supprimer la dernière modification,
- afficher jusqu'à 8 fenêtres à l'écran,
- imprimer le texte en multi-tâches, etc.

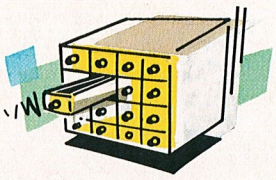
Turbo Editor a été élu "Best of 1985" de la catégorie langages et utilitaires de programmation par Jerry Pournelle de Byte, magazine micro de référence aux USA. Ce dernier suggère à tous les débutants de commencer par le trio Turbo Pascal, Tutor et Editor – avec un minimum de curiosité et d'efforts, vous arriverez vite au stade de programmeur avancé ! (Pour IBM et compatibles).

Turbo Graphix

Tout sur le graphisme et le dessin sur IBM, M24 double résolution, Amstrad 6128, Apricot et cartes Hercules – les routines de Graphix vous permettent de réaliser des applications graphiques de haut niveau :

- gérer les fenêtres graphiques ou texte,
- sauver et lire sur disque des images,
- tracer les lignes, cercles, polygones, les courbes avec lissage, résolution, interpolation,
- zoomer sur une partie d'une fenêtre,
- dessiner des histogrammes, des camemberts

et bien d'autres choses encore !



Turbo Database Toolbox

Contient les outils pour construire une base de données gérée par la méthode des arbres B+, avec multi-critères, pour trier vos fichiers et pour installer vos programmes en Turbo Pascal pour vos utilisateurs. Vous avez le code source d'une petite base de donnée complète sur la disquette. (Disponible pour toutes les machines sur lesquelles tourne Turbo Pascal).

Turbo Gameworks

Trois jeux compilés (pour IBM et compatibles) : échecs, bridge et morpion. Mais Turbo Gameworks est bien plus qu'un nouveau logiciel de jeu : il vous révèle les secrets de la théorie des jeux. En étudiant et modifiant le code source, vous pouvez perfectionner le programme et apprendre à créer vos propres jeux. Ce Toolbox se prête particulièrement bien à la démonstration et à l'enseignement de la programmation. A vous de jouer !

Nouveau !

Voici les adaptations les plus récentes de la ligne Turbo Pascal sur des nouvelles machines :
Turbo Pascal, Tutor, Database Toolbox sur Amstrad PCW, Commodore 128, machines MSX, Einstein – Turbo Graphix sur Amstrad 6128, Apricot émulant l'écran IBM, Sirius, Olivetti double résolution – Turbo Pascal option graphique sur Amstrad 464, 664 et 6128 (si vous désirez échanger votre version de Turbo Pascal pour l'option graphique, renvoyez votre disquette maître à Borland Fraciel, en réglant seulement la différence – voir le bon de commande).
Tous les manuels sont maintenant disponibles en français. Si vous avez renvoyé votre licence d'utilisation, vous pouvez changer gratuitement votre manuel anglais en nous le retournant avec vos coordonnées.

SERVICE PLUS

BORLAND FRACIEL, c'est aussi un ensemble complet de services. Sur simple demande, vous recevrez gratuitement une documentation sur les produits qui vous intéressent.
Notre service technique est à votre disposition pour tout renseignement et conseil de programmation. N'oubliez pas de nous renvoyer votre licence d'utilisation pour pouvoir en profiter !
En outre vous serez régulièrement tenu au courant des nouveautés et des mises à jour concernant les produits Borland.

Et comment vous procurer les produits Borland ?

Vous pouvez commander directement chez Borland Fraciel en renvoyant le bon de commande ci-dessous, par téléphone ou par télex (diverses possibilités de règlement : chèque, contre-remboursement, carte bleue ou mandat administratif (pour les collectivités). Vous trouverez également Turbo Pascal chez les meilleurs revendeurs et dans les FNAC.

CONCEPTION "SCENARISTES ASSOCIES"

BON DE COMMANDE

Règlement joint ☐
Carte Bleue (date d'exp.) ____/____
|____|____|____|____|

Contre-Remboursement
(France uniquement) + 25 F ☐

Nom _____ Prénom _____
Adresse _____
_____ Tél. _____

PORT 15 F TTC PAR PRODUIT

☐ **TURBO Pascal 3,0 MS-DOS, PC-DOS :** 800 F HT (948,80 TTC)
☐ **TURBO Pascal 3,0 CP/M 80 :** 625 F HT (741,25 TTC)
☐ **TURBO Pascal 3,0 option graphique :** 800 F HT (948,80 TTC) pour Amstrad CPC 464, 664 et 6128
☐ **Échange Turbo Pascal pour Turbo Pascal avec option graphique :** renvoyez la disquette originale + 175 F HT (207,55 TTC) (port gratuit)

☐ **TURBO 87 :** 1.350 F HT (1.601,10 TTC)
☐ **TURBO BCD :** 1.350 F HT (1.601,10 TTC)
☐ **TUROB 87 + BCD :** 1.650 F HT (1.956,90 TTC)
☐ **TURBO Tutor :** 350 F HT (415,10 TTC)
☐ **TURBO Graphix Toolbox :** 675 F HT (800,55 TTC)
☐ **TURBO Database Toolbox :** 625 F HT (741,25 TTC)
☐ **TURBO Gameworks Toolbox :** 700 F HT (830,20 TTC)
☐ **TURBO Editor Toolbox :** 700 F HT (830,20 TTC)

ORDINATEUR : _____ Taille de la disquette : _____
Système d'exploitation avec numéro de version : _____

Pour tout renseignement et une documentation gratuite :

BORLAND FRACIEL
Logiciel n'est-ce pas ?

78, rue de Turbigo 75003 PARIS - **Tél. : 1/42.72.25.19 - Télex : 216120**

Les données sont donc recopiées sous forme magnétique sur un disque souple ou dur, ce qui suppose plusieurs étapes. En temps normal, lorsque l'ordinateur n'a pas de lecture / écriture sur disque, les disquettes sont « au repos ».
Sur un *PC* moyen, l'ordinateur lorsqu'il effectue une lecture fichier, va faire tourner le lecteur de disquettes, attendre qu'il atteigne une certaine vitesse, environ 300 tours par minute, puis, positionner la tête de lecture sur la Table d'Allocation des Fichiers, et rechercher où se trouve le fichier requis ; repositionner la tête de lecture sur la piste voulue, attendre que la tête se stabilise sur cette piste, lire l'information, la convertir en charges électriques et finalement la mettre dans la *RAM*.

Si l'application utilisée effectue donc beaucoup d'échanges avec les unités de disque/disquette, un temps assez important pourra être gagné en utilisant un disque virtuel dans la *RAM*, car, dans ce cas, pour une lecture fichier par exemple, c'est un transfert d'électrons d'une partie de la *RAM* dans une autre, celle gérée par le *DOS* qui s'effectuera.

PAS DE PANIQUE...

Le seul danger possible lorsque l'on utilise un disque virtuel, est d'oublier de recopier les données mises à jour dans ce disque simulé dans la *RAM*, dans une véritable disquette ou un véritable disque. Car si l'on éteint le *PC* avant cette recopie, toutes les données *RAM* disparaissent, et, bien entendu, celles du disque virtuel également.

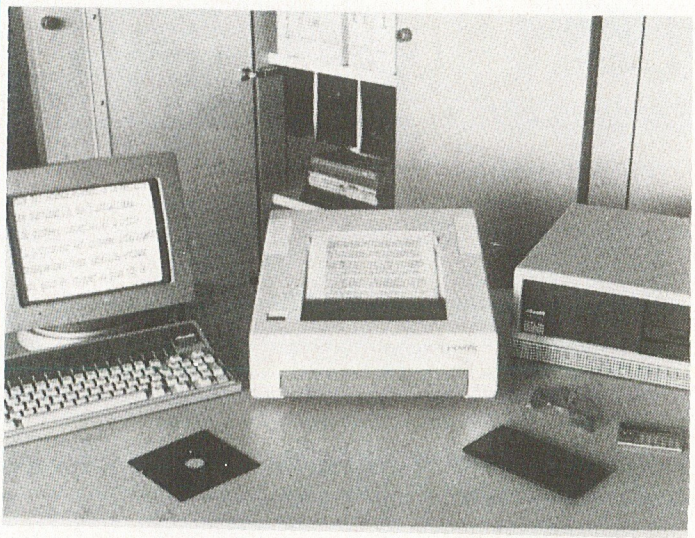
Certains constructeurs de cartes multifonctions ont essayé de résoudre ce problème en permettant d'effectuer cette sauvegarde des fichiers même après que le *PC* ait été éteint malencontreusement, du fait de la présence d'une batterie incorporée qui continue d'alimenter la carte.

C'est donc là un point qui peut se révéler assez important lors de l'achat d'une carte, si toutefois l'utilisateur est plutôt distrait.

EMS

Le disque virtuel va permettre d'effectuer des recopies de la portion de la *RAM* dépassant les 640 K, ou moins adressées par le *DOS*, dans la partie gérée par celui-ci. Si le fichier est assez important dans le disque virtuel, un temps relativement « long » sera tout de même occupé par des copies et recopies à l'intérieur de la *RAM* elle-même, entre la partie *DOS* et la partie disque virtuel.
Vers 1985, une technique d'adressage de la mémoire supplémentaire a donc été mise au point et la spécification *EMS* ou *Expansed Memory Specification est apparu*.
Avec cette technique, l'intégralité de la mémoire dépassant les 640 K est gérée à partir du *DOS*, ce qui élimine, par conséquent, le besoin de définir un disque virtuel, un très grand fichier pouvant être recopié intégralement dans la *RAM* et y être géré.
Cette technique a été supportée par un assez grand nombre de logiciels tels que *SuperCalc 3, Windows, Framework II, 1.2.3, Multiplan 2* etc.

Apprenez à lire à votre ordinateur



SIC

Transforme vos feuilles dactylographiées en fichiers "traitement de texte".

Saisit les articles que vous sélectionnez dans vos revues.

Alimente vos bases de données par lecture d'annuaires, lexiques, tableaux...

Avec le logiciel AP, vous pouvez lui apprendre à reconnaître toute nouvelle police de caractères ou tout symbole graphique.

Vous pouvez aussi saisir vos images et schémas pour les inclure dans vos fichiers "traitement de texte".

Périphérique + Logiciel de base 49 000 F H.T.

Logiciel d'apprentissage AT 30 000 F H.T.

Identification à l'écran d'une lettre à apprendre.

Sélection au feutre d'un article dans une revue.

inovatic

IMMEUBLE LES QUADRANTS
3, av. du Centre
78180 Montigny-le-Bretonneux
Tél. (16-1) 30.57.22.11

Service lecteur n° 005 (dernière page)

Livre de 226 pages et 2 disquettes

69F (TTC)

Cet ouvrage didactique illustré est accompagné du programme Framework II complet, mais sans impression ni sauvegarde. Vous pourrez avec lui vous initier et évaluer Framework II, le standard des logiciels intégrés, qui a été élu logiciel de l'année.
Framework II comprend : tableur, graphique, traitement de texte, fichiers, table des matières et bien d'autres choses... Framework II comporte à la fois le plus grand tableur (32000 lignes × 32000 colonnes) et le traitement de texte le plus complet (correcteur orthographique de 80.000 mots).

FRAMEWORK II, TOUT VOTRE BUREAU DANS UN SEUL LOGICIEL

- **Framework II, l'intégré de seconde génération** Micro-Ordinateurs
- **Framework II le Magnifique** Décision Informatique
- **Framework II : un superbe outil, aujourd'hui parvenu à maturité** Décision Informatique
- **Plus et mieux avec Framework II** Décision Informatique
- **Framework II, la puissance et la convivialité** OPC
- **Framework prend du muscle** Science et Vie Micro
- **Framework II, la seconde version du progiciel intégré le plus vendu** Soft IBM-PC
- **Framework II, plus puissant et plus grande mémoire** OP Magazine
- **Framework II, tout un programme** Le Point

BON DE COMMANDE

à retourner à : La Commande Electronique
7, rue des Prias
27920 Saint-Pierre de Bailleul

M/Mme ..... Prénom : .....

Société : .....

Adresse : .....

.....

Code Postal ..... Ville : .....

Tél : .....

vous commande .... ouvrage(s)
"Prise en main de Framework II" au prix unitaire de 69 F (TTC) franco de port.
Règlement par chèque joint à la commande.

RU

La Commande Electronique

ASHTON-TATE

En fait, la liste s'allonge chaque mois, et, progressivement regroupe les plus célèbres logiciels du marché.
Cependant, la spécification *EMS* n'est pas une panacée.
Sous *Lotus 1.2.3,* par exemple, la gestion de la mémoire étendue se fera pour des données et non pour des formules, ce qui restreint plus ou moins l'utilisation de la mémoire étendue à la base de données, ce qui n'est toutefois pas négligeable. Un assez grand nombre de cartes compatibles *EMS* sont apparues récemment, *Quadboard-AT* de *Quadram* et *Supermax/EMS* de *IDEAssociates* en sont quelques exemples.
Si l'on en croit *Intel,* plus de 100 éditeurs de logiciels seraient en train d'adapter leur produits à la gestion de *EMS.*
Face à *EMS,* nous avons vu apparaître une deuxième tentative de standard, qui bien que supportée par *Ashton Tate,* n'a pas été suivie par les éditeurs de logiciels à ce jour, *EEMS.*
EEMS, qui signifie *Enhanced Expanded Memory Specification,* utilise une technique d'adressage de la mémoire supplémentaire légèrement différente de *EMS,* et, n'est pas gérée comme telle par les logiciels cités plus haut.
EEMS propose, en revanche, de travailler avec plusieurs logiciels en parallèle sous *DOS,* ce qui est appréciable.
Finalement, *Quadram* semble avoir résolu le problème, à sa manière, en développant une carte compatible avec les deux « standards ».
Néanmoins, toutes ces cartes pourront être utilisées dans l'usage traditionnel de disque *RAM.*

VDISK

Pour créer un disque virtuel sous *DOS 3.0* et au-dessus, il faut utiliser l'utilitaire *VDISK.SYS.*
Dans certaines versions de *DOS,* cet utilitaire est appelé *RAMDISK.SYS.*
Pour pouvoir l'utiliser, il faut mettre à jour le fichier *CONFIG.SYS.*
CONFIG.SYS est un fichier qui est lu par le système lorsque l'on allume l'ordinateur. S'il est présent, le système va lire son contenu et prendre en compte ce qui y est indiqué pour « configurer » le *PC.*
Un certain nombre de logiciels tels que *Wordstar 2000* ou *dBASE III Plus* viennent modifier des valeurs dans ce fichier afin de pouvoir tourner.
dBASE III Plus, par exemple, vient indiquer un nombre de fichiers supérieur à celui que le *DOS* gère en standard.
CONFIG.SYS doit donc être présent sur la disque *DOS* qui sert au démarrage, sur un *PC* avec lecteur(s) de disquettes uniquement ou sur le répertoire principal de votre disque dur dans cet autre cas.
Si le fichier *CONFIG.SYS* n'existe pas, il suffit de le créer avec *EDLIN* ou un éditeur de textes similaire.
Pour indiquer que nous voulons créer une unité C: virtuelle avec 128 octets dans un *PC* à deux lecteurs de disquettes, il nous faudrait introduire dans *CONFIG.SYS* la ligne :

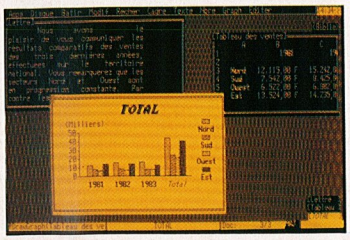
DEVICE = VDISK.SYS 128

Par défaut, le disque virtuel qui est créé comporte des « secteurs » virtuels de 128 octets.
Il faut savoir que la gestion des disquettes virtuelles sera effectuée par le

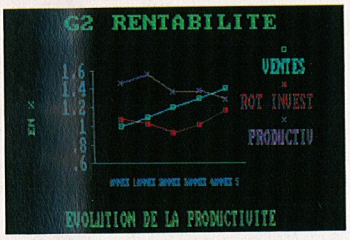
CARTE GRAPHIQUE SOLUTION

3.300 F (HT)

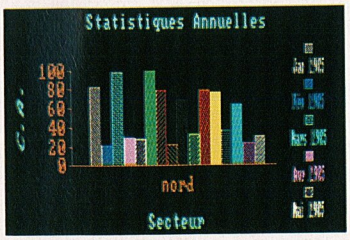
Monochrome



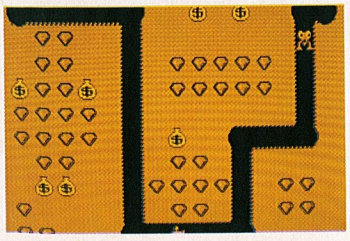
Hercules



Couleur



16 et 8 Couleurs pour
Framework II
Lotus 1-2-3 et Symphony



16 tons de vert
Jeux sur écran monochrome

SOLUTION, UNE SEULE CARTE GRAPHIQUE POUR TOUTES LES FONCTIONS

- MONOCHROME STANDARD
- COMPATIBLE HERCULES
- GRAPHIQUE COULEUR
- 16 ET 8 COULEURS POUR FRAMEWORK II, LOTUS 1-2-3 ET SYMPHONY
- 16 TONS DE VERT. LES GRAPHIQUES COULEUR SUR VOTRE ECRAN MONOCHROME
- LES JEUX SUR VOTRE ECRAN MONOCHROME
- PORTS PARALLELE ET SERIE EN OPTION

La Commande Electronique

DOS comme s'il s'agissait de véritables disquettes, et donc par secteurs.
Nous pouvons indiquer que nous désirons travailler avec des secteurs de 512 octets, ce qui correspond aux secteurs des disquettes sur *PC :*

DEVICE = VDISK.SYS 128 512

Un troisième paramètre permet d'indiquer le nombre de fichiers présents dans la racine du disque virtuel, la valeur par défaut étant 64.
Si nous travaillons, par exemple, avec de tous petits fichiers, nous pourrons monter ce chiffre :

DEVICE = VDISK.SYS 256 128 100

Nous aurions 100 fichiers potentiellement dans les 256 K octets de *RAM* disque, sur des secteurs virtuels de 128 K.
Pour que le système prenne ce disque virtuel en compte, il faudra « *re booter* » le système avec *Ctrl Alt et Del.*

EXTENSION

Enfin, nous allons pouvoir ajouter un paramètre : /E afin d'indiquer au *DOS* de créer le disque virtuel en mémoire « étendue », c'est-à-dire au delà des 640 K adressables par le *Dos* sur une carte de type *EMS.*

DEVICE = VDISK.SYS 1028 512 100 /E

Dans l'exemple ci-dessus, nous allouons plus d'un million d'octets au disque virtuel et le plaçons en mémoire étendue, ce qui fait que l'intégralité de la *RAM* conventionnelle reste disponible pour les programmes.
Après avoir émis l'une quelconque des commandes ci-dessus, laquelle créerait un disque virtuel C: sur un *PC* à deux unités de disquettes, (D: sur une unité à disque dur, etc.), nous pouvons recopier un ou plusieurs fichiers dans le C: virtuel:

COPY B:CLIENTS C:

Pour votre programme, vous travaillerez avec le fichier CLIENTS sur C: et donc avec des temps de réponse très fortement améliorés, et, il suffira au retour du programme de recopier à nouveau CLIENTS sur B:

offre promotionnelle Logicom

20 600 F HT

L'imprimante haut de gamme et son PC/XT

L'IMPRIMANTE : • modèle Dataproducts* SPG 8072 • 400 cps/132 col. • 200 cps qualité lettre • graphique • introduction feuille à feuille • couleur et cassette 200 feuilles en option.
*Dataproducts est une marque déposée de Dataproducts Corp.

LE MICRO : • 128 k RAM • compatible IBM* PC/XT • floppy 360 k/octets • carte graphique • moniteur haute résolution.
*IBM est une marque déposée de International Business Machines.

Logicom (1) 64 46 31 31 Z.A. DE COURTABŒUF AVENUE DE L'ATLANTIQUE 91942 LES ULIS CEDEX

UNIFILE®

LA PUISSANCE EN TOUTE SIMPLICITÉ...

UNIFILE : système de gestion de fichiers et de base de données relationnelle.

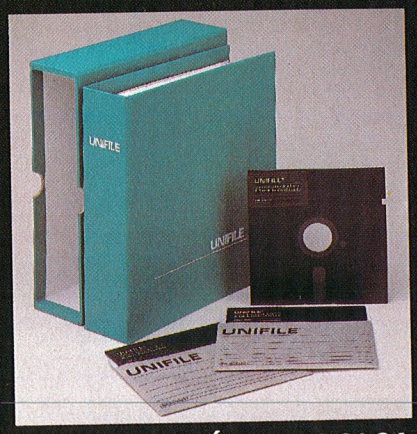
SIMPLICITÉ - Avec UNIFILE rentrez dans le monde de la puissance dans la simplicité. Pas de langage, pas de syntaxe, pas de programme, pas d'acrobatie avec les touches du clavier.

PUISSANCE ET RAPIDITÉ - UNIFILE est d'une remarquable puissance et d'une grande rapidité. Il permet de mettre en œuvre des applications faisant intervenir jusqu'à 9 fichiers simultanément. On peut ainsi effectuer instantanément des transferts de données à travers les fichiers (sans programme et sans syntaxe).

EFFICACITÉ - UNIFILE c'est l'efficacité avant tout. Il est mis en place et opérationnel très rapidement auprès d'utilisateurs non expérimentés sans avoir recours à de longues séances de formation.

PRIX - 4 950 F HT
Y compris disquette contenant de nombreuses applications prêtes à l'emploi et modifiables (fichiers, gestion de stock, salaire).

Fonctionne sur IBM® PC/XT/AT et compatibles sous MS/DOS® 128 ko mini., 192 ko pour utilisation maximum.

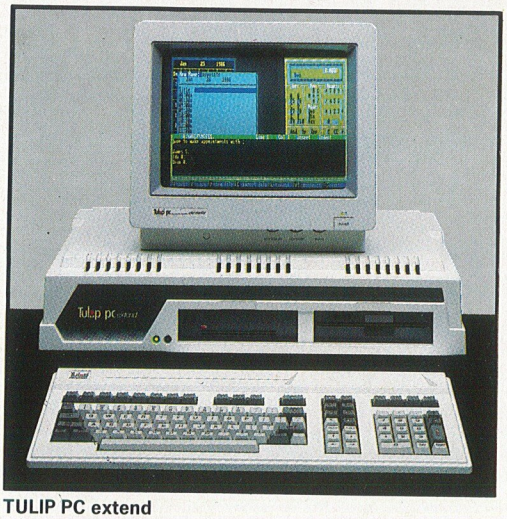


LABSTAR
39, RUE DOMBASLE 75015 PARIS TÉL. 42.50.21.34

ACANTHE

IBM® est une marque déposée de International Business Machines. MS/DOS® est une marque déposée de MICROSOFT.

POUR CHERCHER LE MEILLEUR PC VOUS POUVEZ RESTER EN EUROPE



TULIP PC extend
microprocesseur 8086, fréquence d'horloge 8 MHz. ● co processeur arithméthique (opt.) 8087/8 MHz. ● mémoire centrale (standard) 640 ko ● adaptateur graphique monochrome (standard) 80 × 25, format 720 × 348 ● adaptateur couleur graphique (opt.) 7 couleurs: 40/80 × 25, format 320/640 × 200: monochrome: 80 × 25, format 640 × 200/400 ● jeux de caractères internationaux: 11 ● interfaces: clavier compatible IBM 2; E/S parallèle: compatible Centronics; E/S série: compatible RS 232; contrôleur disques souples: 2 unités; bus d'extension pour cartes d'extension compatible IBM: 3 emplacements; piggy backed slots E/S; crayon lumineux compatible TTL ● horloge temps réel • alimentée par pile ● éprom moniteur 16 ko ● unités disque souple 40 pistes 2 × 360 ko ● unité disque dur (opt.) 1 × 21 mo, 1 × 32 mo ● système d'exploitation (standard) MS-DOS 3.1 ● interpréteur (standard) GWBasic ● Window manager (standard) MS-WINDOWS avec applications MS-paint et MS-write ● unités séparées: bande magnétique, 10 mo + bande magnétique, 32 mo + bande magnétique ● set transportable ●

TULIP PC compact
microprocesseur 8088, fréquence d'horloge 8 MHz. ● coprocesseur arithmétique (opt.) 8087/8 MHz. ● mémoire centrale (standard) 512 ko ● adaptateur graphique monochrome (standard) 80 × 25, format 720 × 348 ● adaptateur graphique couleurs (opt.) 7 couleurs: 40/80 × 25, format 320/640 × 200: monochrome: 80 × 25, format 640 × 200/400 ● jeux de caractères internationaux: 11 ● interfaces: clavier compatible IBM I: E/S parallèle: compatible Centronics; E/S série: compatible RS 232; contrôleur disques souples: 2 unités; bus d'extension pour cartes d'extension compatible IBM: 4 emplacements; piggy backed slots E/S: crayon lumineux compatible TTL ● horloge temps réel ● éprom moniteur 16 ko ● unités disque souple 40 pistes 2 × 360 ko ● unité disque dur (opt.) 1 × 10 mo. ● système d'exploitation (standard) MS-DOS 3.1 ● interpréteur (standard) GWBasic ● Window manager (standard) MS-WINDOWS avec applications MS-paint et MS-write ● unités séparées: bande magnétique, 10 mo + bande magnétique ●

TULIP PC compact:
2 unités disque souple, adaptateur graphique monochrome, mémoire interne 512 ko, clavier et moniteur:
Prix public: **FFRS 14.500,–** h.t.

Tulip® computers

Siège Pays-Bas: TULIP COMPUTERS INTERNATIONAL B.V. Hambakenwetering 2 NL 5231 DC 's-Hertogenbosch. Télephone 31.73.413235, Telex 50316 nl. **Belgique:** Brusselpoortstraat 8, B 2800 Mechelen. Téléphone 32.15.422480, Télex 64698 b. **Allemagne:** Hindenburgstrasse 23, D 4300 Essen 1. Téléphone 49.201.227452.

AMII INFORMATIQUE

163 RUE DE ROME
75 017 PARIS

46.22.22.39. +

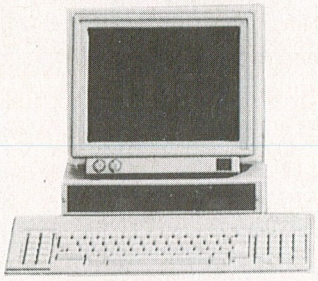
22.39.
¡QUE FACILE

LogAbax
L'ENSEMBLIER INDUSTRIEL INFORMATIQUE.

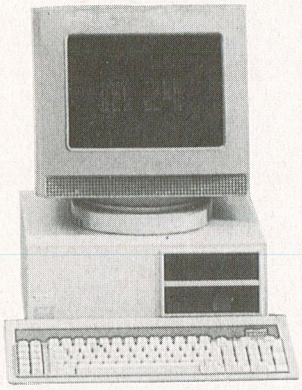
NOUVEAU TEAM MICRO OLIVETTI.

olivetti
LogAbax

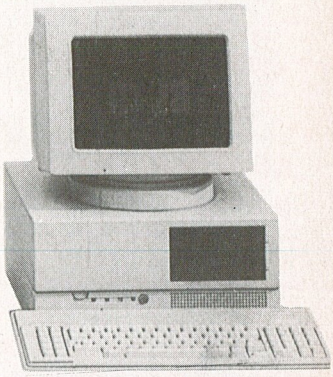
Olivetti, c'est l'esprit Formule 1,



PERS 1300/14690FF.HT



PERS 1600/19590FF.HT



PERS 1800(20Mo)
/40450FF.HT

Comme [redacted] moins cher qu'[redacted]

VICTOR

VICTOR VPC 2

- 640 Ko mémoire de base
- 2 unités de Disquettes 360 Ko
- Interface série parallèle
- Clavier AZERTY
- Ecran monochrome
- Livré avec MS DOS 3.1 et GW - Basic

VPC2 ~~11900-FF/HT~~ 10900 FF/HT

VICTOR V 286

- 512 Ko mémoire de base
- Disque dur 20 Mo
- Disquette 1,2 Mo
- Ecran monochrome
- Clavier AZERTY
- Sortie série parallèle
- Livré avec MS DOS 3.1 et GW - Basic

V286 (20 Mo)
~~35900-FF/HT~~ 34490 FF/HT

UNE PUISSANCE DU TONNERRE!

olivetti
LogAbax

PERS 1200(10Mo)/37930FF.HT

PAR MINITEL

LA COMMANDE PAR MINITEL
7 JOURS SUR 7, 24 H SUR 24
Appelez le réseau Télétel
16 (3) 615.91.77
Tapez connexion Fin
Puis tapez

COMTLE

FAIRE CHOIX :
N° 5 « AMII »

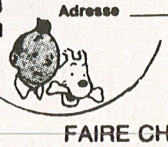
BON DE COMMANDE A RETOURNER A « AMII-INFORMATIQUE »

QUANTITE	DESIGNATION	PRIX UNITAIRE	PRIX TOTAL

MODE DE REGLEMENT ☐ Chèque bancaire joint ☐ C.C.P. joint ☐ Mandat-lettre joint

SOUS TOTAL	
PORT	
Contre REMBOURSEMENT	
TOTAL T.T.C.	

Nom _____ Ville _____
Prénom _____ CP _____
Adresse _____ Date _____



AMII INFORMATIQU
163, RUE DE ROME - 75017 PARIS
(1) 46.22.22.39

l'état de notre périphérique CON: (ANSI.SYS est chargé).

Nous vous rappelons que le témoin de CON: est ouvert en permanence et vaut 00 pour l'entrée standard et 01 pour la sortie standard :

```
A>DEBUG
_A
2D12:0100 MOV BX,0
2D12:0103 MOV AX,4400
2D12:0106 INT 21
2D12:0108
_G 0108

AX=80D3  BX=0000  CX=0000  DX=80D3  SP=CFAE  BP=0000  SI=0000  DI=0000
DS=2D12  ES=2D12  SS=2D12  CS=2D12  IP=0108   NV UP DI PL NZ NA PO NC
2D12:0108 CC            INT     3
_Q
A>
```

Après avoir chargé les registres BX à 0 et AX à 4400, nous trouvons au retour de INT 21 NC dans les indicateurs, c'est à dire Not Carry ; cela s'est donc bien passé.

Le mot d'état se trouve donc dans DX avec la valeur 80D3H, c'est à dire en binaire : 1000 0000 1101 0011, ce qui se lit :

- ☐ Bit 15 = 1,
- ☐ Bit 14 = 0 : IOCTL non supportées,
- ☐ Bit 13 à 7 = 00 0000 1,
- ☐ Bit 6 = 1, pas de fin de fichier rencontrée,
- ☐ Bit 5 = 0, vérifie les caractères de contrôle,
- ☐ Bit 4 = 1, réservé,
- ☐ Bit 3 = 0, Ce n'est pas l'horloge,
- ☐ Bit 2 = 0, ni NUL:,
- ☐ Bit 1 = 1, c'est CON: en sortie,
- ☐ Bit 0 = 1, et en entrée.

Toutes ces informations sont cohérentes avec ce que nous connaissons de CON: et du système.

CONTRÔLE DE PÉRIPHÉRIQUES

Quatre fonctions sont à notre disposition pour communiquer avec le gestionnaire de périphérique, suivant qu'il s'agit d'un périphérique caractères ou bloc, et suivant notre désir d'envoyer des informations de contrôle, ou celui de connaître l'état de ce périphérique.

Il s'agit des fonctions :

- ☐ 4402H envoi d'une chaîne de caractères de contrôle à un périphérique caractères.
- ☐ 4403H reception d'une chaine de caractère de contrôle.
- ☐ 4404H envoi de caractères de contrôle à un périphérique bloc.
- ☐ 4404H réception de caractères de contrôle pour un bloc.

Les registres contiennent les informations suivantes (avant ou après l'interruption numéro 21H) dans le cadre de l'appel de ces fonctions :

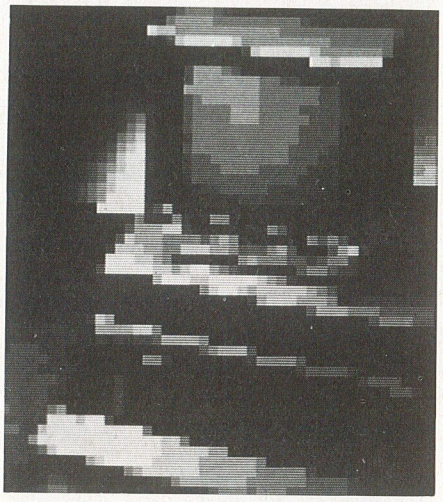
AX : Code de la fonction.
BX : Témoin pour un périphérique caractère, Numéro de périphérique (disquette par défaut = 0, A: = 01, B: = 02, ...)
CX : Nombre d'octets à transférer.
DS:DX : Adresse d'une zone mémoire où transférer ces CX octets.

Si cela se passe bien le Carry revient à zéro et AX contient le nombre d'octet transférés, sinon le Carry revient positionné à 1 et AX contient un des codes d'erreur suivant :

- ☐ 1 : fonction invalide.
- ☐ 5 : disque(tte) invalide.
- ☐ 6 : témoin invalide.

ÉTAT D'IOCTL

La fonction 4406H permet de vérifier l'état en entrée sur un périphérique caractère ou sur un fichier, la fonction 4407H permet d'en vérifier son état en sortie.



Pour cela, il nous suffit de mettre dans BX le témoin du fichier ou périphérique concerné, dans AX le code 4406H ou 4407H, puis de déclencher l'interruption numéro 21H.

Si celà se passe bien, le Carry est à 0 et l'information revient dans le registre AL : 00 pour périphérique pas prêt (ou fichier en fin de fichier), FFH pour périphérique prêt.

Sinon le Carry revient positionné à 1, et AX contient un des codes d'erreur suivant :

- ☐ 1 : valeur invalide dans AL,
- ☐ 5 : accès refusé,
- ☐ 6 : témoin invalide dans BX, ou
- ☐ D : données invalides.

Voyons un exemple de ceci en demandant au DOS sous DEBUG l' état en sortie de CON: (témoin 0001) et en entrée de PRN: (témoin 0003) :

L'informatique professionnelle pour tous avec les "PC-X" de MICROFORCE LD

UNE GAMME DU PC * A L'AT * GARANTIE UN AN
DÉPANNAGE EN 72 HEURES

PC-XA 256 KO 1 LECTEUR 360 KO **6 900** F HT

PC-XB 256 KO 2 LECTEURS 360 KO **7 900** F HT *

PC-X1 640 KO 2 LECTEURS 360 KO **8 900** F HT

PC-X2 640 KO 1 LECTEUR 360 KO DISQUE DUR 20 MO **15 900** F HT

PC-ATX 6 systèmes

MICROFORCE

NOUS CONSULTER

NOS PRIX INCLUENT L'UNITÉ CENTRALE SELON DESCRIPTIF

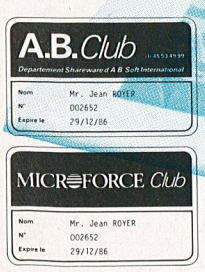
CARTE UNITÉ CENTRALE 8 SLOTS D'EXTENSION

— Processeur 8088, 4,77 MZ
— 256 KO mémoire extensible A 640 KO
— 640 KO pour PC-X1 et X21
— 1 carte multifonctions horloge, calendrier, port parallèle, port série, sortie jeux contrôleur de disquettes
— 1 carte graphique couleur et monochrome ou graphique monochrome haute résolution

Les PC-X1 et X2 sont livrés avec carte graphique multifonctions
— 1 clavier azerty compatible
— 1 système d'exploitation MS/DOS 2.11 avec licence manuel MS-DOS en français
— 1 documentation hardware complète
— 1 alimentation 135 W
— 1 boîtier métallique rigide à charnière

Tous nos systèmes sont évolutifs et peuvent supporter 4 lecteurs - Compatibilité 3ème niveau

Livraison forfaitaire 160 F HT sur toute la France.



Nous vous offrons la première année d'abonnement aux clubs

Le meilleur club de logiciels presque gratuits en prime la première disquette de logiciels

Seulement dans le cadre du club, l'accès au catalogue de produits professionnel à des prix «CLUB» software et hardware.

EXTRAIT DU CATALOGUE CLUB :

— Écran monochrome vidéo composite ..... 950 F HT
— Écran 14" monochrome haute résolution ..... 1 700 F HT
— Écran 12" monochrome haute résolution ..... 1 200 F HT
— Écran 14" graphique couleur 3 fonctions : couleur, mono. vert mono. ambre ..... 3 900 F HT
— Imprimante matricielle 100 cps 80 cl ..... 2 700 F HT
— Imprimante matricielle 100 cps 132 cl ..... 3 490 F HT
— Souris PC avec carte d'adaptation ..... 990 F HT
— Disquettes DFDD 100% certifiées (10) ..... 140 F HT
— Nombreuses cartes d'extention nous consulter

N'HÉSITEZ PAS A NOUS APPELER UNE ÉQUIPE DE PROFESSIONNELS VOUS RÉPONDRA ET VOUS CONSEILLERA AVEC PLAISIR.

* Marque déposée IBM. CORP.

3, rue Michel Carré
95100 ARGENTEUIL

Tél. 39.61.87.25
39.61.35.36

Télex : ATN 695 758 F

Pour plus d'informations, cerclez 172
Service lecteur télématique p. 133

```
A>DEBUG
_A
2D12:0100 MOV BX,1
2D12:0103 MOV AX,4407
2D12:0106 INT 21
2D12:0108
_G 108

AX=01FF  BX=0001  CX=0000  DX=0000  SP=CFAE  BP=0000  SI=0000  DI=0000
DS=2D12  ES=2D12  SS=2D12  CS=2D12  IP=0108   NV UP DI PL NZ NA PO NC
2D12:0108 CC            INT     3
_
_A 100
2D12:0100 MOV BX,3
2D12:0103 MOV AX,4406
2D12:0106 INT 21
2D12:0108
_G=100 108

AX=0000  BX=0003  CX=0000  DX=0000  SP=CFAE  BP=0000  SI=0000  DI=0000
DS=2D12  ES=2D12  SS=2D12  CS=2D12  IP=0108   NV UP DI PL NZ NA PO NC
2D12:0108 CC            INT     3
_Q
A>
```

A chaque fois, INT 21 nous a répondu NC comme état du Carry, celà s'est donc bien passé, et nous lisons AL = FFH c'est à dire 'Clavier prêt' et AL = 00, c'est à dire 'Imprimante pas prête' (Effectivement, nous n'avons aucune imprimante d'installée sur notre PC).

SUPPORT AMOVIBLE ?

La série de fonction d'IOCTL que nous allons maintenant aborder ne sont disponibles qu'avec **une version 3.0 du DOS**.

La fonction 4408H nous permet de vérifier si nous avons affaire à un lecteur de disque(tte) permettant le changement ou non du (de la) disque(tte).

Pour celà, il nous suffit de mettre dans BL le numéro de drive (0, 1 pour A:, 2 pour B:, ...) et 4408 dans AX.

Après l'interruption numéro 21H, si celà s'est bien passé, le Carry revient à 0 et AX contient 1 pour un disque fixe, ou 0 pour une unité disquette (changement de disquette possible).

Si par contre celà s'est mal passé, le Carry revient à 1 et AX contient le code d'erreur 0001 si le « device » ne supporte pas cette fonction, ou 000E si le numéro du disque(tte) est invalide.

Voyons sous DEBUG un exemple de cette fonction pour le lecteur de disquette A: et pour notre disque virtuel C: :

```
A>DEBUG
_A 107
2D12:0100 MOV BL,1
2D12:0102 MOV AX,4408
2D12:0105 INT 21
2D12:0107
_G

AX=0000  BX=0001  CX=0000  DX=0000  SP=CFAE  BP=0000  SI=0000  DI=0000
DS=2D12  ES=2D12  SS=2D12  CS=2D12  IP=0107   NV UP DI PL NZ NA PO NC
2D12:0107 CC            INT     3
-
```

Le nouveau standard statistique pour PC

Le logiciel SAS/STAT™ de SAS Institute

Offrez à votre micro-ordinateur la puissance des fonctions statistiques SAS® : avec le logiciel SAS/STAT, un logiciel intégré répondant à vos besoins statistiques sur PC.
Les fonctionnalités SAS sur PC sont les mêmes que celles sur site central ou mini-ordinateur. De plus, une interface utilisateur complètement interactive s'appuyant sur le concept de fenêtre permet d'entrer en relation avec les différentes parties de la session SAS en cours : programme, sorties, suivi de l'exécution. Et un lien micro-site central vous permet de choisir **où** et **comment** utiliser le Système SAS*.
Vous voulez gérer des données d'expérimentation de manière quotidienne ?
Faire une analyse de variance sur des plans équilibrés ou non équilibrés ?
Construire des modèles linéaires ? Le logiciel SAS/STAT répond à toutes ces demandes et à bien d'autres.

Régression/Analyse de variance

- sept types de sélection des modèles
- statistiques d'influence, diagnostics de collinéarité, courbes de régression partielles, valeurs calculées, résiduelles, et limites de confiance
- somme des carrés, statistiques de Fisher, et probalités sauvegardées dans des tableaux
- addition et suppression interactives de variables et d'observations
- possibilités d'analyse des plans à mesures répétées
- large choix de méthodes de comparaisons multiples des moyennes
- possibilité de choisir des tests d'hypothèses sur une ou plusieurs variables
- estimation de paramètres en utilisant un algorythme QR optionnel

Analyse multivariée et analyse discriminante

- en composantes principales
- analyse factorielle avec plusieurs méthodes d'estimation y compris le maximum de vraisemblance
- corrélations canoniques
- fonctions discriminantes linéaires ou quadratiques
- classification de tableaux de données en appliquant les fonctions discriminantes
- nombreuses statistiques et estimations en sortie

* Le logiciel SAS/STAT tourne sur IBM PC, PC AT, PC XT et compatibles disposant d'un disque dur et d'au moins 512 K de mémoire. L'accès aux fichiers DBF (dBase II® et dBase III®) ainsi qu'aux fichiers DIF est fourni. Pour utiliser le logiciel SAS/STAT, vous avez besoin du logiciel SAS de base pour micro-ordinateur.

* SAS est une marque déposée de SAS Institute INC., Cary, NC, USA. SAS/STAT est une marque de SAS Institute Inc. Un commentaire doit accompagner la première mention de chaque marque déposée ou de chaque marque et doit préciser que chaque marque référencée est utilisée pour identifier les produits et services de SAS Institute Inc., dBase II et dBase III sont les marques déposées d'Ashton-Tate, Culver City, CA.
Copyright© 1986 par SAS Institute

Le logiciel SAS/STAT complète le logiciel SAS de base sur micro-ordinateur. Ainsi, vous bénéficiez de la puissance du langage SAS pour vos calculs :

- création de nombres pseudo-aléatoires pour les distributions normale, binomiale, gamma, de Poisson et autres
- probabilités et quantiles pour de nombreuses distributions standard y compris les distributions de Student, Fisher et du Chi-2

De plus, le logiciel SAS de base sur PC offre un large éventail d'outils pour la manipulation des données... et des fonctionnalités statistiques simples pour les calculs de fréquence, moyennes, etc.

Un service complet sur PC

SAS Institute vous offre une gamme d'outils statistiques fiables et puissants... mais aussi une formation et de la documentation. Notre prestation comprend également le support technique, les mises à jour et une période d'essai gratuite de 45 jours.
Installez le logiciel SAS/STAT sur votre micro-ordinateur et appréciez ses capacités. Retournez le coupon-réponse ou appelez notre Département Marketing dès aujourd'hui pour plus de précisions sur nos outils statistiques sur micro-ordinateur.

SAS Institute s.a.
50, avenue Daumesnil
75579 PARIS CEDEX 12
Tél. : (1) 43 42 54 63
Télex : 216545 SASFR
Télécop. : (1) 43 47 34 64

☐ Je désire recevoir une documentation sur le logiciel SAS/STAT

☐ Je désire avoir un contact avec votre service Marketing

Veuillez compléter et nous renvoyer ce coupon

Mr Mme ..... Fonction.....

Organisme .....

Adresse .....

Code postal..... Ville .....

Téléphone .....

SAS®
SAS Institute
50, avenue Daumesnil - 75579 PARIS CEDEX 12

```
_A100
2D12:0100 MOV BL,3
2D12:0102
_G=100 107

AX=0001  BX=0003  CX=0000  DX=0000  SP=CFAE  BP=0000  SI=0000  DI=0000
DS=2D12  ES=2D12  SS=2D12  CS=2D12  IP=0107   NV UP DI PL NZ NA PO CY
2D12:0107 CC            INT     3
_Q
A>
```

Pour A:, le Carry revient à 0, et nous voyons dans AX (0000) que la disquette est changeable.

Par contre, pour C:, nous voyons (CY) que le « device » RAMDRV0.SYS ne supporte pas les IOCTL.

TENTATIVE D'ACCÈS EN 'FICHIERS PARTAGÉS'

Le DOS 3.0 permet l'utilisation par plusieurs programme de même fichier, moyennant le chargement de système de partage de fichiers.

Lorsqu'une tentative d'accès à un fichier échoue lorsque par exemple un autre programme s'en est réservé l'accès, le DOS réitère un certain nombre de fois cette tentative d'accès à intervalle régulier.

La fonction 440BH permet de modifier le nombre d'essais (dans BX) et le délais (dans CX).

Profitons en ici pour mentionner l'exeistance de 2 fonctions du DOS 3.0 que nous avions passées sous silence à propos de la gestion de fichier, les fonctions 5C00H et 5C01H permettant à un processus de se réserver l'accès exclusif à une partie d'un fichier, et de libérer une partie ainsi réservée ; les autres processus tentant d'accéder à ce fichier échoueront.

GESTION DU RESEAU MICRO-SOFT

Un réseau est le moyen de connecter plusieurs PC entre eux, et de leur faire partager des ressources communes.

Nous n'avons pas ici l'intention de rentrer dans chacune des fonctions du DOS 3.0 permettant de gérer ce réseau (à moins qu'un grand nombre d'entre vous nous le demande) ; ceci déborderait du cadre que nous nous étions fixé dans cette série d'article, nous allons cependant récapituler les diverses possibilités que nous offre le DOS 3.0.

Périphérique « bloc » sur le réseau ? : la fonction 4409H vérifie si la lettre de référence à un périphérique (C:, D:, ...) nous renvoies sur le réseau, ou fait partie de notre PC local.

Fichier sur le réseau ? : La fonction 440AH vérifie si le fichier auquel il est fait référence par un témoin appartient au réseau ou au PC local.

Nom de notre PC pour le réseau ? : La fonction 5E00H nous retourne le nom sous lequel le réseau reconnaît notre PC ainsi qu'un numéro d'identification.

Initialisation de l'imprimante : La fonction 5E02H nous permet d'ajouter un en-tête à tout lot de données destinée à une imprimante situé sur le réseau. Ceci permet à chaque station du réseau de reconfigurer l'imprimante à son gout.

Gestion de la liste de matériel et fichier sur le réseau : La fonction 5FH, possède 3 sous fonctions (5F02H, 5F03H et 5F04H) permettnt de consulter la liste des ressources disponibles sur le réseau, d'en ajouter ou d'en retrancher.

TABLEAUX RÉCAPITULATIFS

Liste des fonctions

00H : Fin de programme — Revue numéro 22.
01H : Saisie et echo d'un caractère — Revue numéro 19.
02H : Affichage d'un caractère — Revue numéro 18.
03H : Entrée sur terminal auxiliaire (liaison série) — non traité.
04H : Sortie sur terminal auxiliaire — non traité.
05H : Impression d'un caractère (imprimante) — non traité.
06H : Entrée-sortie directe sur la console — Revue numéro 18 et 19.
07H : Attente et lecture directe d'un caractère — Revue numéro 19.
08H : Saisie d'un caractère — Revue numéro 19.
09H : Affichage d'une chaine de ceractère — Revue numéro 19.
0AH : Edition d'une chaine de caractères — Revue numéro 19.
0BH : Etat du clavier — Revue numéro 19.
0CH : Destruction de la mémoire tampon du clavier puis saisie — Revue numéro 19.
0DH : Réinitialisation des disques — Revue numéro 23.
0EH : Changement de disque — Revue numéro 23.
0FH à 17F : Gestion de fichier (DOS 1.0) — non traité.
18H : Réservée.

19H : Disque en cours — Revue numéro 23.
1AH : Changement de l'adresse de la zone de transfert — Revue numéro 23.
1BH : Données sur le disque courrant — Revue numéro 23.
1CH : Données sur un disque — Revue numéro 23.
1DH à 20H : Réservées.

FICHE TECHNIQUE VTX-MAN

VTX-MAN outil d'aide au développement d'applications Vidéotex..

1. LOGICIEL VTX-MAN

Fonctions de définitions des pages Vidéotex

Cette fonction constitue un puissant outil d'AIDE AU DÉVELOPPEMENT d'applications vidéotex.

- Composition et modification de pages vidéotex :
  (Normes Télétel et graphique)

- Définition des enchaînements de pages (Arborescence)
  . Sur action des touches de fonctions des minitels
  . Sur codes d'enchaînements de type «MENU»
  . Sur mots clés ou mnémoniques.

- Définition des mots de passe

- Impression des pages vidéotex

- Intégration de pages VIDÉOTEX : (normes télétel) venant d'applications extérieures.

Gestion des données

- Définition des zones variables
  Les zones variables de saisie ou d'affichage des MINITELS ouvrent une orientation vers les applications spécifiques développées en BASIC, PASCAL, C, et tous langages (APL - D.B. III ...) et permettent l'exploitation de tous fichiers sous DOS.

VTX-MAN prend en charge les contrôles élémentaires.

. Longueur de la zone variable.
. Type de caractères (Alpha ou numérique)
. Nombre de décimales.

Fonction d'éxécution

Moniteur transactionnel

Cette fonction réalise la mise en œuvre opérationnelle des applications réalisées.

VTX-MAN est un logiciel OUVERT permettant aux possesseurs d'IBM PC : XT, AT ou compatibles, d'utiliser leur micro-ordinateur en SERVEUR VIDÉOTEX.

VTX-MAN EST LE LIEN ENTRE TOUS LES FICHIERS MS/DOS ET LEUR EXPLOITATION EN MODE VIDÉOTEX.

VTX-MAN

. Gestion des terminaux Minitels
. Service d'accueil et contrôle des mots de passe
. Gestion des pages vidéotex et de leurs enchaînements
. Activation des transactions spécifiques
. Gestion des temps de connexion
. Déconnexion automatique sur TIMER

Journal cyclique

Cette fonction offre la possibilité de distribuer un journal cyclique qui sera offert à tout minitel venant se connecter simultanément à la fonction serveur.

Modularité

VTX-MAN s'adapte aux besoins propres de chaque utilisateur lui permettant de procéder aux extensions nécessaires au développement des capacités du Micro-Serveur.

. De 1 à 32 accès simultanés
. Jusqu'à 3275 pages vidéotex
. Jusqu'à 10 000 mots clés ou mnémoniques
. Jusqu'à 36 variables par page
. Jusqu'à 30 mots de passe
. Paramétrage des temporisations pour la connexion (de 20 secondes à 3 heures)
. Paramétrage de la voie en composition
. Répartition des voies d'accès entre TRANSPAC et réseau Télécommuté (fil à fil et intercom)
. Compatibilité Réseau.

2. LES MATÉRIELS ASSOCIÉS A

Carte de communication

- COM 1 (1 voie)
- PC TRANS 20 (4 voies intelligentes).

Émulation : Minitel.

- PC TRANS 33 (carte - Modem intégré)

Modems (RTC).

- Modems 1200/75 bauds
- Modem URT (sitintel) de 4 à 32 voies.

Coffret europad (transpac)

- OST 4 à 32 voies
- URX (sitintel) 16 ou 32 voies.

3. LES SERVICES ISA-VTX-MAN

- Formation VTX-MAN (1 journée)
- SAV : assistance au logiciel. Maintenance matériel
- Développement : installation clé en main d'applications spécifiques sur micro-vidéotex. (Nombreuses références)
- Réseau de revendeurs agréés sur toute la France.
- Messageries - (Aurelia - Isabel)

IMP SIPAG 64 58 22 05

21H à 25H : Gestion de fichier (DOS 1.0) — non traité.
25H : Change un vecteur d'interruption — Revue numéro 22.
26H à 28H : Gestion de processus et fichiers (DOS 1.0) — non traité.
29H : Extraction d'un nom de fichier — Revue numéro 23.
2AH-2BH : Saisie-modification de la date — Revue numéro 23.
2CH-2DH : Saisie-modification de l'heure — Revue numéro 23.
2EH : VERIFY — Revue numéro 23.
2FH : Obtient l'adresse de la zone de transfert — Revue numéro 23.
30H : Numéro de version du DOS — Revue numéro 23.
31H : Reste résident — Revue numéro 22.
32H : Réservée.
33H : BREAK — Revue numéro 23.
34H : Réservée.
35H : Lecture d'un vecteur d'interruption — Revue numéro 22.
36H : Place sur le disque — Revue numéro 23.
37H : Réservée.
38H : Informations nationnales — Revue numéro 23.
39H : Création de répertoire — Revue numéro 20.
3AH : Effacement de répertoire — Revue numéro 20.
3BH : Changement de répertoire — Revue numéro 20.
3CH : Création de fichier — Revue numéro 21.
3DH : Ouverture de fichier — Revue numéro 21.
3EH : Fermeture de fichier — Revue numéro 21.
3FH : Lecture de fichier — Revue numéro 21.
40H : Ecriture de fichier — Revue numéro 21.
41H : Destruction de fichier — Revue numéro 20.
42H : Change le pointeur dans un fichier — Revue numéro 21.
43H : Attribut de fichier — Revue numéro 20.
44H : IOCTL — Revue numéro 25.
45H - 46H : Manipulation de témoin de fichier — Revue numéro 21.
47H : Nom du répertoire en cours — Revue numéro 20.
48H : Demande de mémoire — Revue numéro 22.
49H : Libération de mémoire— Revue numéro 22.
4AH : Changement de taille mémoire allouée — Revue numéro 22.
4BH : Chargement de programme (et overlay) — Revue numéro 22.
4CH : Fin de processus — Revue numéro 22.
4DH : Code de retour d'un processus — Revue numéro 22.
4EH - 4FH : — Recherche dans un répertoire — Revue numéro 20.
50H - 54H : Réservées.
55H : VERIFY — Revue numéro 23.
56H : Renomme un fichier — Revue numéro 20.
57H : Date et heure d'écriture dans un fichier — Revue numéro 21.
58H : Stratégie d'allocation mémoire — Non traitée.
59H : Code d'erreur étendu — non traitée.
5AH : Création de fichier temporaire — Revue numéro 21.
5BH : Création de nouveau fichier — Revue numéro 21.
5CH : Verrouillage d'accès à une partie de fichier — Revue numéro 25.
5DH : Réservée.
5EH - 5FH : Réseau Micro-Soft — Revue numéro 25.
60H à 61H : Réservées.
62H : Adresse du préfixe de segment de programme — Revue numéro 22.
63H à 7FH : Réservées.

Codes d'erreur

1 : Code de fonction invalide.
2 : Fichier pas trouvé.
3 : Chemin pas trouvé.
4 : Trop de fichiers ouverts.
5 : Accès refusé.
6 : Témoin invalide.
7 : Bloc de contrôle mémoire détruit.
8 : Pas assez de mémoire.
9 : Adresse invalide de bloc de mémoire.
10 : Environnemment invalide.
11 : Format invalide.
12 : Code d'accès invalide.
13 : Données invalides
14 : Réservé.
15 : Lecteur invalide.
16 : Tentative de destruction du répertoire courrant.
17 : Ce n'est pas le même lecteur.
18 : Plus de fichier.
19 : Disque protégé en écriture.
20 : Mauvaise unité de disque.
21 : Lecteur pas prêt.
22 : Commande Disque invalide.
23 : Erreur de CRC.
24 : Longueur invalide (opération disque).
25 : Erreur de recherche.
26 : Ce n'est pas un disque MS DOS.
27 : Secteur pas trouvé.
28 : Plus de papier.
29 : Défaut d'écriture (disque).
30 : Défaut de lecture.
31 : Problème général.
32 : Violation de partage de ressource.
33 : Accès à un fichier protégé.
34 : Mauvais disque.
35 : Mauvais FCB.
36 à 49 : Réservés.
50 : Pas de réseau.
51 à 72 : Erreur sur la gestion du réseau.
73 à 79 : Réservés.
80 : Fichier éxistant.
81 : Réservé.
82 à 88 : Réseau.

Bibliographie ayant servie à cette étude :

« Micro-Soft MS-DOS, Operating system, Programmer's Reference » (classeur Micro-Soft en anglais).
« Guide du PC-DOS » de Richard Allen King chez Sybex.
« DOS Technical Reference », IBM (en anglais).
« Programmation du 8086-8088 » de James W. Coffron chez Sybex.
« Au coeur de l'IBM PC, logique et fonctionnement interne » de J.B. Thièle chez édi TESTS.
« Assembleur 8088 et BIOS IBM PC » de Bruno Le Maire et Gilles Mauffrey chez édi TESTS.

Nous terminons ici, avec des tableaux récapitulatifs, notre exploration des fonctions du DOS, celle-ci nous a permis d'en apprendre énormeement sur le DOS et sur ce qu'est un système (et nous l'espérons vous l'a permis aussi).

Si cependant, il y a des points sur ce sujet sur lesquels vous désireriez plus d'informations ou que vous aimeriez voir repris de manière plus approffondie n'hésitez pas à écrire à la Revue.

Signalons, avant de mettre le point final, le démarrage d'une nouvelle série, baptisée « Adresses utiles », qui va vous permettre de découvrir où se trouve des informations, telles que, l'état du haut-parleur, du clavier etc.

■

LE PASCAL
Déclarations structurées et fichiers

Dans notre précédent article, nous avons commencé à vous présenter le Pascal, et vous donner un aperçu de ses possibilités. Avant de vous présenter toutes sa puissance, au niveau structure de programme et sous-programme, nous allons aujourd'hui nous attacher à quelques déclarations structurées supplémentaires et aux fichiers.

Jean-Paul PRUNIAUX.

Nous avons déjà mis l'accent sur l'application de la programmation structurée au Pascal, sur la structure générale du programme, et sur les instructions structurées. Celle-ci s'applique aussi au niveau de la définition des variables pour que nous puissions mettre en forme nos variables de travail, nos tableaux, etc. de manière à coller au plus prés, au problème que nous voulons traiter.
Nous vous rappelons ici qu'une structure est un ensemble organisé et ordonné d'éléments.

DE NOUVEAUX TYPES DE DONNEES

Nous avons vu, dans notre précédent article, que nous avions un certain nombre de types de base que reconnaît le Pascal (Entier, Reel, Caractère, Logique, Octet). En fait le Pascal nous laisse la possibilité de décrire tout autre type de données dont nous pourrions avoir besoin, déclarations que nous pourrons utiliser ultérieurement dans les déclarations CONST et VAR par exemple.

Une variable est en fait définie par la liste des valeurs ou des états que nous désirons lui vouloir prendre. Un choix arbitraire, mais pratique existe déjà pour les types de base, par exemple un entier prend les valeurs -32 768 à 32 767, et pour les caractères, la suite des possibilités de votre clavier.

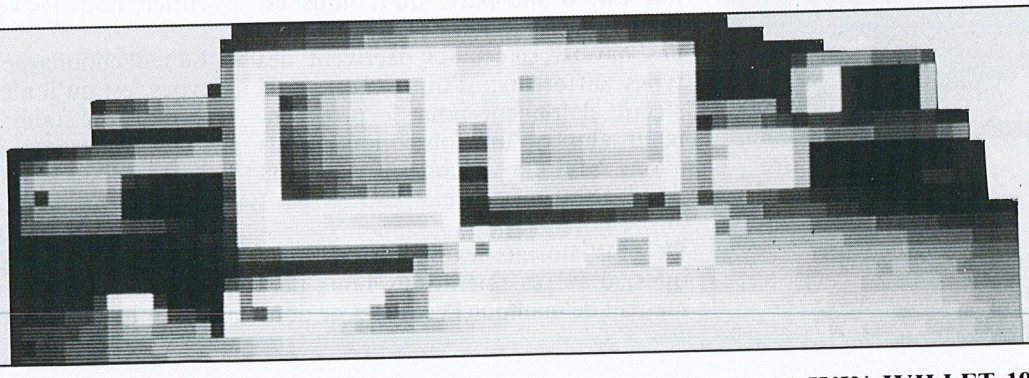
Nous pouvons définir un nouveau type de variable, soit en énumérant la liste des états que peut prendre cette variable, soit en précisant un intervalle de valeurs d'un type déjà éxistant.

Description des valeurs

Considérons par exemple que nous voulions travailler avec les jours de la semaine. Nous pouvons définir un nouveau type de variable que nous allons baptiser Jour et qui prendra les états : Lundi, Mardi, Mercredi, ... Dimanche, de la manière suivante :

```
TYPE
Jour = (Lundi, Mardi, Mercredi,
Jeudi, Vendredi, Samedi, Dimanche) ;
```

Il faut voir ici que les éléments Lundi, Mardi, ... sont internes au Pascal, et ne correspondent en aucun cas au chaîne de caractères 'Lundi', 'Mardi', ... Ce n'est qu'une liste d'états, au même titre que le type Boolean possède les 2 états True et False. Voyons en un exemple :



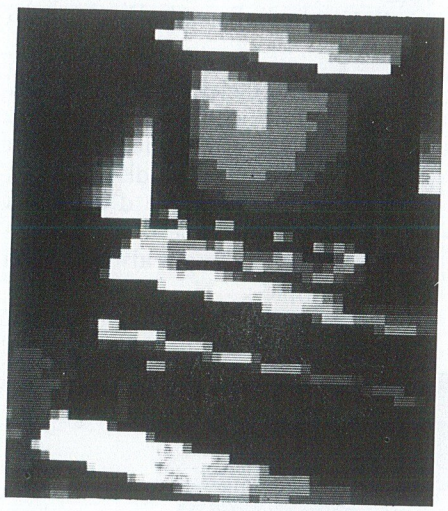
```
PROGRAM Type0;
TYPE
   Jour = (Lun, Mar, Mer, Jeu, Ven, Sam, Dim);

VAR
   Day : Jour;
   Msg : ARRAY [Jour]  OF  STRING [8];

BEGIN
   Msg [Lun] := 'Lundi';        Msg [Mar] := 'Mardi';
   Msg [Mer] := 'Mercredi';     Msg [Jeu] := 'Jeudi';
   Msg [Ven] := 'Vendredi';     Msg [Sam] := 'Samedi';
   Msg [Dim] := 'Dimanche';

   FOR Day := Lun TO Dim DO
      WRITE (Msg[Day], ' ');
END.
```

```
Running
Lundi Mardi Mercredi Jeudi Vendredi Samedi Dimanche
```

Nous commençons ici par déclarer un nouveau type de variable pouvant prendre les valeurs Lun à Dim (les jours de la semaine). Ceci nous permet de déclarer une variable Day de ce type, ainsi qu'un tableau de message. Ce tableau comprend autant de chaîne de 8 caractères que nous avons déclaré de jours. Le seul indice possible pour accéder à ce tableau doit être de type jour.

Comme notre liste de jour est ordonnée, nous pouvons faire une boucle dessus, et même des tests :

```
IF Jour <= Ven THEN
   WRITE ('Jour ouvrable');

DO CASE Jour OF
Lun : WRITE ('Debut de la semaine');
Sam : WRITE ('Samedi');
Dim : WRITE ('Repos');
END;
```



L'avantage énorme de ce type de donnée est, d'une part, qu'il nous est impossible d'ajouter des carottes avec des navets, si ceux-ci décrivent des types différents, d'autre part, ceci s'écrit clairement pour le programmeur, alors qu'une fois compilé, nous avons à faire à un code prenant peu de place en mémoire.

De plus, hormis les opérations logiques, d'autres opérations nous permettent de manipuler ces types de donnée, entre elles ou de les convertir en entier, nous les verrons plus loin.

Le seul ennui avec ces déclarations de types, est qu'il n'existe pas de moyen de dialogue standard avec le clavier et l'écran, charge au programmeur de les écrire s'il en a besoin. Pour en assurer l'impression, nous avons eu recours à un tableau de chaînes de caractères que nous aurions pu déclarer sous forme constante plus rationnellemnt de la manière suivante :

```
PROGRAM Type0;
TYPE
   Jour = (Lun, Mar, Mer, Jeu, Ven, Sam, Dim);

CONST
   Msg  : ARRAY [Jour] OF STRING [8]
        = ('Lundi',     'Mardi',  'Mercredi', 'Jeudi',
           'Vendredi', 'Samedi', 'Dimanche');
VAR
   Day : Jour;

BEGIN

   FOR Day := Lun TO Dim DO
      WRITE (Msg[Day], ' ');
END.
```

Intervalle

Une autre manière de définir un nouveau type de données consiste à décrire un intervalle de valeurs faisant partie d'un type déjà existant, permettant ainsi de construire des sous-ensembles de type existant :

TYPE
Plage—Nb = 1..10 ;
Week—End = Sam..Dim ;

Un exemple d'application de ces déclarations de type peut être trouvé dans le domaine des jeux, pour représenter l'état d'une grille sur laquelle nous pourrions avoir des pions blancs, des pions noirs, ou pas de pion du tout, avec :

TYPE
Plage = 1..10 ;
Case = (Vide, Noir, Blanc) ;
VAR
Grille : ARRAY [Plage,Plage] OF Case ;

Ceci nous définit une grille de 10 par 10 dans laquelle nous pouvons ne rien avoir ou un pion blanc ou noir.

Opérations sur ces données typées

Nous avons déjà vu que les opérations logiques sont disponibles sur ces variables typées, et des exemples avec un IF et un CASE, nous avons les autres fonctions suivantes, toutes basées sur le fait que ces données typées sont ordonnées :
SUCC (Var) donne une variable du même type que Var qui la succède dans la liste.
PRED (Var) donne une variable de même type que Var qui la précède dans la liste.
ORD (Var) donne un entier qui correspond au rang de la variable dans cette liste.

Voyons un l'exemple en reprenant nos déclarations des jours de la semaine, puis, grâce à un boucle Repeat Until en visualisant toutes ces fonctions pour chaque données possibles de type jour. Remarquez cependant que le prédécesseur de Lundi n'existe pas, le successeur de dimanche non plus, nous en affichons donc l'état des fonctions concernant ces valeurs en dehors de la boucle.

```
PROGRAM Type0;
TYPE
   Jour = (Lun, Mar, Mer, Jeu, Ven, Sam, Dim);

CONST
   Msg  : ARRAY [Jour] OF STRING [8]
        = ('Lundi',     'Mardi',  'Mercredi', 'Jeudi',
           'Vendredi', 'Samedi', 'Dimanche');
VAR
   Day, Day_P, Day_S : Jour;

BEGIN
   WRITELN ('Jour':9, ' Ord', 'Pred':10, 'Succ':10);
   Day := Lun;
   WRITELN (Msg [Day]:9,          ORD (Day):4,
            '???':10,             Msg [SUCC (Day)]:10);

   Day := SUCC (Day);
   REPEAT
      WRITELN (Msg [Day]:9,          ORD (Day):4,
               Msg [PRED (Day)]:10, Msg [SUCC (Day)]:10);
      Day :=  SUCC(Day);
      UNTIL Day = Dim;

   WRITELN (Msg [Day]:9,          ORD (Day):4,
            Msg [PRED (Day)]:10, '???':10);
END.
```

```
Running
      Jour Ord       Pred       Succ
     Lundi   0        ???      Mardi
     Mardi   1      Lundi   Mercredi
  Mercredi   2      Mardi      Jeudi
     Jeudi   3   Mercredi   Vendredi
  Vendredi   4      Jeudi     Samedi
    Samedi   5   Vendredi   Dimanche
  Dimanche   6     Samedi        ???
```

Pour les variables de type caractères, Turbo-Pascal met à notre disposition la fonction CHR(), similaire au CHR$ du Basic permettant de transformer un nombre entier en un caractère, en respectant la table des codes ASCII.

Autre extension du Pascal standard, le Turbo-Pascal nous permet de convertir des données d'un type à l'autre. En reprenant nos déclarations précédentes, la ligne suivante transforme un nombre en jour :

Day := Jour (3) ;

Nous trouverons alors dans jour, la valeur correspondant à Jeudi, la transformation se fait en respectant la valeur de ORD.

STRUCTURE DE DONNÉES

Comme premiers exemples de données structurées, nous avons vu dans notre précédent article les tableaux et les chaînes de caractères avec les déclarations ARRAY et STRING. Bien que ces possibilités soient nécessaires, elles ne nous apportent pas grand chose de plus par rapport aux autres langages. En Pascal, nous avons une autre déclaration à notre disposition, RECORD, nous permettant d'organiser un ensemble de données dans une structure.

Supposons que nous désirions travailler avec un fichier de personnes, et que nous voulions considérer la fiche de base comprenant : nom, prénom, adresse, nous pourrions utiliser la déclaration suivante :

```
PROGRAM Record1;
TYPE
   Genre = (Nul, Mr, Mme, Mle);
   Carte = RECORD
           Sexe : Genre;
           Nom  : STRING [20];    Pre  : STRING [20];
           Ad1  : STRING [40];    Ad2  : STRING [40];
           END;
```

Après avoir défini un type de données Genre, permettant de distinger monsieur de madame, le bloc Record, End ; nous permet de définir un nouveau type de données obéissant à la structure décrite à l'intérieur de celui-ci.

Nous y trouvons le code du genre, puis le nom, prénom et adresse sous forme de chaîne de caractères. Nous pouvons maintenant déclarer deux variables fiche1 et fiche2 avec ce type carte, c'est à dire deux fiches. (Auparavant, nous en profitons pour décrire en constantes les chaînes de caractères nous permettant de visualiser le genre de chaque personne) :

```
CONST
   M_Genre : ARRAY [Genre] OF STRING [4]
           = ('', 'Mr ', 'Mme ', 'Mle ');
VAR
   Fiche1, Fiche2  : Carte;
```

Ces fiches étant créées, nous avons plusieurs moyens de les manipuler. Nous pouvons considérer une case d'une fiche, ou bien la fiche en totalité.

Pour s'adresser à une case d'une fiche, il nous faut mettre le nom de la fiche, suivi d'un point et du nom de la case concernée. Voyons les exemples suivants où l'expression Fiche1.Sexe, permet de remplir la première information de Fiche1, puis fiche1.Nom, la deuxième, et ainsi de suite.

L'expresssion Fiche2 := Fiche1 ; permet de recopier tout le contenu de la fiche1 dans la fiche2.

Nous ne pouvons traduire directement à l'affichage le contenu de fiche2, aussi nous devons afficher les cases de cette fiche une par une ; cependant, nous verrons, quand nous en serons à la gestion de fichiers que nous n'aurons à spécifier que le nom de la fiche pour l'y écrire.

```
BEGIN
   Fiche1.Sexe := Mr;
   Fiche1.Nom  := 'DUPONT';                Fiche1.Pre := 'Pierre';
   Fiche1.Ad1  := '10 rue du Cerisier'; Fiche1.ad2 := '13000 Marseille';

   Fiche2 := Fiche1;

   WRITELN (M_Genre [Fiche2.Sexe],        Fiche2.Nom, ' ', Fiche2.Pre);
   WRITELN (Fiche2.Ad1);                  WRITELN (Fiche2.Ad2);
END.
```

```
Running
Mr DUPONT Pierre
10 rue du Cerisier
13000 Marseille
```

Comme à l'écriture, il peut être lourd de répéter ces préfixes Fiche1. ou Fiche2. alors que nous ne sommes en train de travailler que sur la même fiche, il existe une solution avec l'instruction With permettant de préfixer une instruction (ou un bloc Begin, End ;) avec ce nom de fiche. (A notre avis cela doit aussi simplifier le code généré par le compilateur). Cette instruction With obéit à la syntaxe suivante :

WITH Variable DO Instruction ;

Reprenons le code de notre programme ci-dessus, et modifions-le en conséquence :

```
BEGIN
   WITH Fiche1 DO BEGIN
      Sexe := Mr;
      Nom  := 'DUPONT';                Pre := 'Pierre';
      Ad1  := '10 rue du Cerisier'; Ad2 := '13000 Marseille'
      END;

   Fiche2 := Fiche1;

   WITH Fiche2 DO BEGIN
      WRITELN (M_Genre [Sexe], Nom, ' ', Pre); WRITELN (Ad1); WRITELN (Ad2)
      END
END.
```

Nous pouvons aussi conserver plusieurs Fiches de type Carte en en faisant un tableau en mémoire de la manière suivante :

VAR
Boitier : ARRAY [1..100] OF Carte ;

Nous allons voir maintenant une autre application de la déclaration Record.

FICHIER DE DONNÉES

Un fichier de données, pour le Pascal, est un ensemble numéroté de données,

toutes du même type, en quantité quelconque, stockées sur un disque. Du point de vue du programmeur Pascal, cela se présente en fait comme un nouveau type de variable : File Of ...

Supposons que nous voulions un fichier de données dans lequel nous désirerions stocker des entiers, nous pourrions le déclarer de la manière suivante :

VAR
Fichier : FILE OF INTEGER ;

Le fichier étant déclaré, nous devons, avant de pouvoir l'utiliser, rajouter quelques instructions dans notre programme, car nous avons affaire avec des accès disque. Il s'agit des ouvertures ou créations de fichiers similaires aux instructions OPEN ou CLOSE du Basic.

La première chose que nous allons avoir à faire est de déclarer au système que nous allons utiliser un fichier dont le nom suit par l'instruction Assign, de syntaxe suivante :

ASSIGN (Variable de type file, Chaîne de caractère) ;

Par exemple, l'instruction ASSIGN (Fichier, 'TOTO.DAT') ; affecte à la variable de type File que nous avons déclarée le fichier TOTO.DAT.

Cette affectation de fichier à la variable de type File étant faite, 2 cas peuvent se présenter. Soit le fichier existe et nous voulons l'utiliser tel quel, soit il n'existe pas, ou nous voulons l'écraser (c'est-à-dire réécrire desssus).

Nous devons aussi signaler cette intention au système. Pour cela, lors d'une première utilisation ou pour écraser le fichier, nous allons utiliser l'ordre REWRITE (Variable de type fichier).

Dès lors, nous allons pouvoir à commencer à écrire dans ce fichier et allons utiliser l'ordre Write de syntaxe suivante :

WRITE (Variable type fichier, Liste de variables de type compatible avec le fichier) ;

Avant de terminer le programme, nous devons signaler au système de faire toutes les mises à jour d'écriture sur ce fichier par un ordre Close (Variable de type fichier) ;. En effet lors des accès en écriture, le système de stockage temporaire de toutes les informations à écrire dans une zone mémoire qui lui est propre, de manière à gagner du temps en ne faisant qu'un accès disquette pour écrire plusieurs données. Close force la remise à jour du fichier et le rend disponible pour les autres utilisateurs ou programmes qui vont en avoir besoin.

Reprenons notre fichier d'entiers, et supposons que nous voulions y écrire les entiers de 65 à 90, le programme s'écrit simplement de la manière suivante :

```
PROGRAM Fic1;
VAR
   i : INTEGER;
   f : FILE OF INTEGER;

BEGIN
   ASSIGN  (f, 'INT');
   REWRITE (f);

   FOR i := 65 TO 90 DO
       WRITE (f,i);

   CLOSE (f)

END.
```

```
Running
```

Maintenant que nous avons écrit ce fichier, cherchons à le relire. Pour cela nous allons écrire un autre bout de programme, similaire à celui que nous avons écrit pour la création de ce fichier, mais nous allons devoir apporter les modifications suivantes :

Après le Assign, au lieu d'utiliser l'ordre Rewrite de création de fichier, nous allons utiliser l'ordre Reset () ; nous ouvrant le fichier et nous positionnant à son début.

Au lieu de l'ordre WRITE() ; nous allons utiliser un ordre de lecture de même syntaxe dont le nom est READ.

Signalons ici aussi une fonction logique du Pascal nous permettant de savoir s'il y a encore des données dans notre fichier. Il s'agit de la fonction EOF (Variable de type fichier) ; qui nous dira True si nous sommes en fin de fichier et False dans le cas contraire. (EOF signifie en fait, fin de fichier ou End Of File).

Notre exemple de relecture de ce fichier vient tout seul :

```
PROGRAM Fic2;
VAR
   i : INTEGER;
   f : FILE OF INTEGER;

BEGIN
   ASSIGN (f, 'INT');
   RESET  (f);

   WHILE NOT EOF(f) DO BEGIN
       READ  (f,i);
       WRITE (i:4)
       END;

   CLOSE (f)

END.
```

```
Running
   65  66  67  68  69  70  71  72  73  74  75  76 77  78  79  80  81  82  83
   84  85  86  87  88  89  90
```

Une remarque est à faire, quant à la nature des informations écrites, sur le disque, en effet, le Pascal, contrairement à d'autres langages comme le Basic et même comme le Fortran inscrit ses informations sur le disque en code binaire, et non pas sous forme de caractères.

Ceci est un avantage car cela nous fait des fichiers plus compacts et des temps d'accès disque plus courts.

Le Basic, pour un programme similaire, aurait écrit sur le disque le code des informations suivantes : 'Espace', le chiffre '6', le chiffre '5', 'Retour en début de ligne' puis 'passage à la ligne' pour le chiffre 65. Le pascal, lui, écrit brutalement le code binaire d'un entier valant 65.

Nous allons faire une démonstration en relisant notre fichier avec un type différent, le type caractères.

Les entiers positifs sont codés de la manière suivante sur 2 octets (un octet est une valeur comprise entre 0 et 255) : Premier octet + 256 * Deuxième octet. Ainsi 65 est codé 0,65 en binaire, 66 0,66 et ainsi de suite.

Les caractères sont codés suivant la table des codes ASCII sur un octet de la manière suivante : 'A' = 65, 'B' = 66, ... 'Z' = 90. Le code 0 correspond au caractère nul qui n'est à priori pas affiché.

Reprenons notre programme Fic2 et modifions les déclarations concernant les entiers, en déclarations concernant les caractères :

```
PROGRAM Fic3;
VAR
   c : CHAR;
   f : FILE OF CHAR;

BEGIN
   ASSIGN (f, 'INT');
   RESET  (f);

   WHILE NOT EOF(f) DO BEGIN
       READ  (f,c);
       WRITE (c:2)
       END;

   CLOSE (f)

END.
```

```
Running
 A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S
 T   U   V   W   X   Y   Z
```

Le premier 65 a été affiché comme 'espace','A', le nul qui le suit a été affiché 'espace','espace' et ainsi de suite.

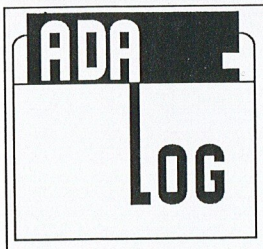
Recherche d'un donnée

Nous vous avons déjà signalé que les informations, à l'intérieur d'un fichier étaient numérotée. La première donnée porte le numéro 0, la deuxième le numéro 1 et ainsi de suite. Le Pascal tient à jour un compteur dans lequel il a le numéro de l'information qui sera concernée par le prochain accès disque. Par exemple, après un Rewrite ou un Reset, ce compteur est à 0. Après un Read ou un Write ce compteur est incrémenté d'une unité.

A tout moment, nous pouvons connaître l'état de ce compteur grâce à la fonction FilePos (Variable de type fichier). Reprenons notre programme Fic2 et faisons lui afficher l'état de ce compteur avant qu'il ne lise le prochain entier :

```
PROGRAM Fic4;
VAR
   i : INTEGER;
   f : FILE OF INTEGER;

BEGIN
   ASSIGN (f, 'INT');
   RESET  (f);

   WHILE NOT EOF(f) DO BEGIN
       WRITE (FILEPOS (f):5);
       READ  (f,i);
       WRITE (i:3)
       END;

   CLOSE (f)

END.
```

```
Running
    0 65    1 66    2 67    3 68    4 69    5 70    6 71    7 72    8 73
    9 74   10 75   11 76   12 77   13 78   14 79   15 80   16 81   17 82
   18 83   19 84   20 85   21 86   22 87   23 88   24 89   25 90
```

Ces logiciels sont vendus par correspondance uniquement.
Disponibles sur IBM/PC et compatibles.
Pour plus de renseignements, appelez le catalogue AdaLog: (1) 45 39 09 99
15 rue Friant
75014 Paris

190F. TTC. Port inclus

Les logiciels professionnels à prix grand public

MODIFICH
Inspectez et modifiez tous vos fichiers en plein écran!

DACTYL'AID
Ne regardez plus votre clavier!

MACHINE/PC
Transforme votre PC+ imprimante en machine à écrire électronique

VISITRAP
Comprendre les appels système sans assembleur!

CLASSIFICHE
La gestion de fiches la plus simple à part le papier et le crayon!

-Uniquement par correspondance
Je désire (indiquer le nombre): ... CLASSIFICHE ... DACTYL'AID ... MACHINE/PC ... MODIFICH ...VISITRAP. Je joins 190F x ... par chèque à l'ordre d'AdaLog.
Nom:..... Adresse: .....

Nous avons bien la donnée numéro 0 contenant la valeur 65, puis la 1 contenant 66, et ainsi de suite.

Nous pouvons modifier la valeur du compteur FilePos() grâce à l'ordre Seek (Cherche) qui obéit à la syntaxe suivante :

SEEK (Variable fichier, Valeur) ;

Cette ordre va manipuler le fichier de manière à ce que le prochain accès se fasse sur la donnée de numéro concerné (dans Valeur). Reprenons l'exemple précédent et faisons lui lire de la dixième information, à la vintième, en ne faisant qu'une lecture sur deux. Attention, la dixième information porte le numéro 9 (Nous avions commencé à compter à 0).

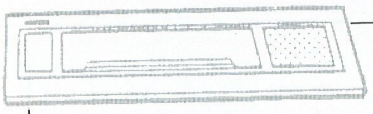
```
PROGRAM Fic5;
VAR
   i,j : INTEGER;
   f   : FILE OF INTEGER;

BEGIN
   ASSIGN (f, 'INT');
   RESET  (f);

   j := 9;
   REPEAT
      SEEK  (f,j);
      WRITE (FILEPOS (f):5);
      READ  (f,i);
      WRITE (i:3);
      j := j + 2
      UNTIL j > 19;

   CLOSE (f)

END.
```

```
Running
    9 74   11 76   13 78   15 80   17 82   19 84
```

En reprenant nos listes de contenu de fichier, nous pouvons vérifier que le dixième élément contient bien 74, la lecture se fait au pas de 2.
Une dernière fonction nous permet de gérer efficacement le compteur d'enregistrement : FileSize (Varfichier) ; nous donnant la taille du fichier en question. Il est à noter que SEEK (Fichier, Filesize(Fichier)) ; nous positionne après le dernier enregistrement, et qu'une écriture faite à cet endroit ajoute un élément au fichier. Affichons grâce à cette fonction la taille de notre fichier INT :

```
PROGRAM Fic2;
VAR
   f : FILE OF INTEGER;

BEGIN
   ASSIGN (f, 'INT');
   RESET  (f);
   WRITE  ('Le fichier INT contient ',FILESIZE(f),' entiers.');
   CLOSE (f)
END.
```

```
Running
Le fichier INT contient 26 entiers.
```



PASCAL

Nous venons ici de terminer le tour des fichiers de données, il est temps de voir un programme d'application nous permettant de récapituler tout ceci.

UN PROGRAMME

Nous allons ici envisager un programme simple de gestion de noms et d'adresses nous permettant de récapituler toutes les notions vues jusqu'à présent en gérant des structures de données de type suivant :

```
TYPE
   Genre = (Nul, Mr, Mme, Mle);
   Carte = RECORD                                  (* Descrition d'une fiche      *)
           Sexe : Genre;                                          { Mr, Mme, Mle    }
           Nom  : STRING [20];    Pre  : STRING [20];             { Nom, Prenom     }
           Ad1  : STRING [40];    Ad2  : STRING [40];             { Adresse         }
           END;
```

Il s'agira de cartes, comme nous pourrions en avoir dans le fichier d'une secrétaire qui comporte une indication Mr, Mme ou Mle, un nom, un prénom et une adresse sur 2 lignes.

Nous allons gérer un fichier de cartes de ce type, et pour cela conserver en mémoire constamment une fiche de travail.

Après les initialisations, dans une grande boucle de travail, notre programme va constamment afficher la fiche de travail, puis nous demander ce que nous désirons en faire.

Au moment de l'affichage de cette fiche de travail, nous visualisons le compteur FilePos() ; remarquez cependant que l'état de ce compteur, est celui après une opération disque, et pointe en fait la fiche suivante. Cela aura le mérite d'être plus parlant pour l'utilisateur de ce programme qui verra un compteur évoluer entre 1 et FileSize() ; qu'entre 0 et FileSize()-1.

Les possibilités de ce que nous pouvons faire de cette fiche sont visualisées sur une ligne de menu « Edit, Lit, Précédent,... » est la réponse de l'opérateur est lue dans la variable Code.

Avant l'affichage de la prochaine fiche, la commande contenue dans Code, est traitée par un Case Code Of de la manière suivante :

Edit : Nous permet de ressaisir complètement la fiche de travail.
Lit : Nous permet si nous ne sommes pas en fin de fichier de lire la fiche suivante.
Précédente : Nous permet si nous ne sommes pas en début de fichier de lire la fiche précédente. Pour cela, il nous faut remonter dans le fichier de la dernière fiche lue, en plus, de la fiche précédente.
Modifie : nous fait modifier la fiche dans le fichier en réécrivant notre fiche de travail dans celui-ci.
Ajoute : nous fait rajouter la fiche de travail en fin de fichier.
Début : nous ramène au début du fichier et nous y fait lire la première fiche.
Cherche : Nous demande un numéro de fiche valide, puis en fait la recherche et la lecture.
Imprime : Nous positionne au début du fichier, puis lit toutes les fiches en séquence et les imprime.

ADRESSEZ-NOUS VOS PROGRAMMES... VOS TRUCS...

POUR PUBLICATION

à la Revue de l'Utilisateur de l'IBM-PC
2, rue d'Amsterdam 75009 Paris

```
PROGRAM Essai_Fichier;
TYPE
   Genre = (Nul, Mr, Mme, Mle);
   Carte = RECORD                                  (* Descrition d'une fiche      *)
           Sexe : Genre;                                        { Mr, Mme, Mle    }
           Nom  : STRING [20];   Pre  : STRING [20];            { Nom, Prenom     }
           Ad1  : STRING [40];   Ad2  : STRING [40];            { Adresse         }
           END;

CONST                                              (* Decodage du genre           *)
   M_Genre : ARRAY [Genre] OF STRING [4]
           = ('', 'Mr ', 'Mme ', 'Mle ');
VAR
   NomFich : STRING [12];                          (* Nom du fichier de travail   *)
   Fiche   : Carte;                                (* Fiche en memoire            *)
   Fichier : FILE OF Carte;                        (* Fichier                     *)
   Code, c : CHAR;                                 (* Reponses de l'operateur     *)
   i       : INTEGER;                              (* Reponse  de l'operateur     *)

BEGIN

(* Initialisations *)
(*-----------------*)

WRITE  ('Nom du fichier: '); READLN (NomFich);          (* Nom du fichier ?  *)
ASSIGN (Fichier, NomFich);                              (* sa declaration    *)

WRITE  (NomFich,' est-il a creer (o/n) ? ');            (* Creation ?        *)
REPEAT READ (KBD, c)
   UNTIL c IN ['O', 'o', 'N', 'n'];

CASE c OF
'o','O'  : BEGIN WRITELN ('Creation');                  (* Oui : Le creer    *)
                 REWRITE (Fichier)
                 END;
'n','N'  : BEGIN WRITELN ('Ouverture');                 (* Non : L'ouvrir    *)
                 RESET   (Fichier)
                 END;
END;

WITH Fiche DO BEGIN                                     (* Mise a zero de la *)
   Sexe := Nul;  Nom := '';  Pre := '';                    { fiche de travail }
   Ad1 := '';  Ad2 := ''
   END;
Code  := 'L';                                           (* Pour tentative de *)
                                                           { lecture d'une f. }

(* Boucle de traitement du code de commande *)
(*------------------------------------------*)

REPEAT
   CASE Code OF

'E'     : WITH Fiche DO BEGIN                           (* Edition de fiche  *)
            WRITELN ('0, 1 Mr, 2 Mme, 3 Mle');             { Saisie d'un code }
            REPEAT READ (KBD,c)                            { pour le genre    }
```

```
                UNTIL c IN ['0'..'3'];                    { entre '0' et '3' }
              Sexe := Genre (ORD (c) - $30);              {convertit en genre}

              WRITE   ('Nom: ');          READ (Nom);     { Saisie du nom    }
              WRITE   (',  Prenom: ');  READLN (Pre);     { Saisie du prenom }
              WRITELN ('Adresse:');
              READLN  (Ad1);              READLN (Ad2);   { 2 lignes adresse }
              END;

'L'     : IF NOT EOF (Fichier) THEN                       (* Lecture d'une fic.*)
              READ  (Fichier, Fiche);

'P'     : IF FILEPOS (Fichier) >1 THEN BEGIN              (* Lecture f. preced.*)
              SEEK  (Fichier, FILEPOS (Fichier)-2);       { Remonte 2 fiches }
              READ  (Fichier, Fiche)
              END
            ELSE WRITELN ('Pas de fiche precedente');

'M'     : IF FILEPOS (Fichier) >0 THEN BEGIN              (* Modifie la fiche  *)
              SEEK  (Fichier, FILEPOS (Fichier)-1);       { Rembobine 1 fiche}
              WRITE (Fichier, Fiche)                      { L'ecrit          }
              END
            ELSE WRITELN ('Pas de fiche existante');

'A'     : BEGIN                                           (* Ajoute le fiche   *)
              SEEK  (Fichier, FILESIZE (Fichier));        { Va fin de fichier}
              WRITE (Fichier, Fiche);                     { Ecriture fiche   }
              END;

'D'     : BEGIN                                           (* Debut de fichier  *)
              SEEK (Fichier,0);                           { Va en debut de f.}
              IF NOT EOF (Fichier) THEN
                 READ  (Fichier, Fiche);                  { Lecture de fiche }
              END;

'C'     : IF FILESIZE (Fichier) > 0 THEN BEGIN            (* Cherche un numero *)
              REPEAT                                      { Demande le numero}
                 WRITE
                  ('Entrez un numero de fiche entre 1 et ',
                  FILESIZE (Fichier), ': ');
                 READLN (i)
                 UNTIL  (i >= 1) AND (i <= FILESIZE (Fichier));

              SEEK (Fichier, i-1);                        { Cherche la fiche }
              READ  (Fichier, Fiche);                     { La lit           }
              END
            ELSE WRITELN ('Pas de fiche existante');

'I'     : BEGIN                                           (* Impression        *)
              SEEK (Fichier, 0);                          { Retour au debut  }
              WHILE NOT EOF (Fichier) DO BEGIN            { Boucle de lecture}
                 READ (Fichier, Fiche);                     { Lecture          }
                 WITH  Fiche  DO BEGIN                      { Impression       }
                    WRITELN (LST, M_Genre [Sexe], Nom, ' ', Pre);
                    WRITELN (LST, Ad1); WRITELN (LST, Ad2); WRITELN (LST);
                    END;
```

```
                END;
            END

ELSE    WRITELN; WRITELN ('Commande inconnue')
END;


(* Affichage et saisie du code *)
(*-----------------------------*)

{ Affichage de la fiche }

   WRITELN; WRITELN
   ('Fiche numero: ',FILEPOS(Fichier));
   WITH Fiche DO BEGIN
      WRITELN (M_Genre [Sexe], Nom, ' ', Pre); WRITELN (Ad1); WRITELN (Ad2);
      END;


{ Menu des commandes possibles }

   WRITELN ; WRITELN
   ('Edit, Lit, Precedent, Modifie, Ajout, Debut, Cherche, Imprime, Fin');

   READ (KBD, Code);
   Code := CHR ( ORD (Code) AND $DF);

   UNTIL Code = 'F';                    (*  FIN DE BOUCLE DE COMMANDE *)
                                        (*-----------------------------*)
CLOSE (Fichier)
END.
```

A l'exécution, nous allons obtenir quelque chose du genre :

```
Edit, Lit, Precedent, Modifie, Ajout, Debut, Cherche, Imprime, Fin

Fiche numero: 2

Mme Durand Jeannette
22 rue des Pleurs
33000 Bordeaux

Edit, Lit, Precedent, Modifie, Ajout, Debut, Cherche, Imprime, Fin

Fiche numero: 3
Mlle Asatourmonte Marguerite
33 rue des Pierres
4900 Angers

Edit, Lit, Precedent, Modifie, Ajout, Debut, Cherche, Imprime, Fin
```

FICHIERS TEXTE

Nous venons de voir les fichiers de données, avec leur découpage en champs de longueur constante, et le stockage des informations sur le disque en code binaire ; nous avons aussi une autre possibilité, plus agréable de gestion de fichier, quand nous gérons du texte, il s'agit des fichiers de texte.

Ceci va compléter ce que nous avons dit dans notre précédent article dans le paragraphe sur les entrées-sorties.

La déclaration d'un tel fichier, identifié avec Fich sera la suivante :

VAR
Fich : Text ;

Dans nos instructions, il faudra l'affecter par un Assign(Fich,) ; puis le créer ou l'ouvrir par un ReWrite(Fich) ; ou un ReSet(Fich) ; avant de l'utiliser et faire un Close(Fich) ; pour sortir du programme, comme avec les fichiers de données précédemment vus.

Ici, nous allons retrouver, entre autre, les instructions vues dans notre précédent article, quant aux entrées-sorties sur la console ainsi que les possibilités de formatage appliquées à un fichier, à savoir :

READ(Fich,...) ; WRITE(Fich,...) ; READLN(Fich,...) ; WRITELN(Fich,...) ;

Reprenons notre programme Fic1 pour lui faire écrire les nombres 65 à 90 sous forme de chiffres et non plus sous forme de codes binaires et visualisons ce fichier grâce à la commande TYPE du DOS :

```
PROGRAM Txt1;
VAR
   i : INTEGER;
   f : TEXT;

BEGIN
   ASSIGN  (f, 'CHIFFRE');
   REWRITE (f);

   FOR i := 65 TO 90 DO
       WRITE (f,i:4);

   CLOSE (f)

END.

A>TYPE CHIFFRE
  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84
  85  86  87  88  89  90
A>
```

De même, nous pouvons faire un petit programme de visualisation, d'un fichier à l'écran, de la manière suivante :

```
PROGRAM Txt2;
VAR
   Ligne : STRING [255];
   Fich  : Text;

BEGIN
   WRITE  ('Nom du fichier a lister: ');
   READLN (Ligne);
   ASSIGN (Fich, Ligne);
   RESET  (Fich);
   WHILE NOT EOF(Fich) DO
      BEGIN
      READLN  (Fich, Ligne);
      WRITELN (Ligne)
      END
END.
```

```
Running
Nom du fichier a lister: AUTOEXEC.BAT
KEYBFR
ECHO OFF
CLS
DATE
TIME
VER
>
```

Signalons ici que nous avons quelques fonctions supplémentaires nous permettant de savoir si nous sommes en fin de ligne, ou en fin de fichier (en ignorant les derniers espaces) :

EOLn() ; SeekEOln() ; SeekEOF() ;.

Nous en avons ici terminer avec le principal sur les fichiers Pascal et Turbo-Pascal, avec un exemple d'application plus proche des préoccupations du gestionnaire (petit fichier d'adresse). Dans notre dernier article notre exemple se situait dans le cadre des jeux, nous allons nous rapprocher de ce domaine prochainement en traitant des sous-programmes Pascal et de leur structure avec des exemples plus proches du technicien, de l'ingénieur ou du mathématicien, à bientôt.

MIEUX UTILISER VOTRE IBM-PC OU VOTRE COMPATIBLE

Dos - dBase - Lotus - Multiplan - Pascal -C - Basic

Tout ce qu'il faut savoir sur le Monde des PC se trouve dans

En vente chaque semaine chez votre marchand de journaux 11 F
Abonnement 395 F à Softin Communications
2, rue d'Amsterdam 75009 Paris

Sauvegarde et Impression d'Ecrans Graphiques sur IMB-PC, en TURBO PASCAL

Graphisme 2D, Graphisme 3D, Jeux, DAO, CAO, traitement d'images, quelque soit le sujet, nous devons faire face à la nécessité de fixer ces images fugitives. Sur papier ou sur disquette, les deux solutions s'avèrent indispensables.
Les utilitaires présentés ici satisferont les besoins des amateurs, chaque jour plus nombreux, de TURBO PASCAL.

Jean-Pierre CHARLES

Aussi nous nous proposons de définir des fenêtres texte en mode graphique, d'étudier les caractéristiques du fichier écran en mémoire centrale, de recopier le fichier écran sur disquette, de la rappeler en mémoire centrale, et de l'imprimer.

MODES GRAPHIQUES SUR IBM PC

TURBO donne accès à trois modes graphiques :
GraphColorMode ; en couleur avec 320 × 200 points,
GraphMode ; en noir et blanc : 320 × 200 points,
HiRes ; en noir et une couleur : 640 × 200 points.
C'est ce dernier mode que nous utiliserons.
Les coordonnées du coin supérieur gauche de l'écran sont 0,0. Les graphiques peuvent être mélangés avec du texte 80 × 25 caractères.
La couleur utilisée est donnée par Couleur, une impression entière entre 0 et 15, de la procédure standard : HiResColor(Couleur) ;
Le retour au mode texte est réalisé par la procédure : TextMode ; appelée en fin de programme graphique.
Toute partie de l'écran peut être définie comme une fenêtre active en mode texte par la procédure :
Window (X1, Y1, X2, Y2) ;
Les X et Y sont les coordonnées absolues des coins opposés de la fenêtre : 1,1,80,25 pour tout 80 colonnes.
Après définition, la fenêtre se comporte comme un écran complet dont le coin supérieur gauche a pour coordonnées 1,1.
GraphWindow (X1, Y1, X2, Y2) ; définit une partie de l'écran comme fenêtre active dans le mode graphique choisi.
En HiRes ; la fenêtre graphique par défaut est : 0,0,639,199.

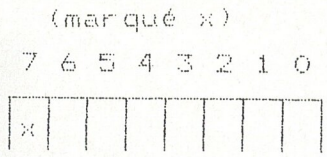
FICHIER MEMOIRE DE L'ECRAN

Mode HiRes, 640 × 200 'PELs'. (PEL : picture élément).
Tous les points graphiques sont adressables.

La mémoire tampon d'affichage (Display Buffer), commence à l'adresse 'B80000' et occupe 16 kO de mémoire. Elle est organisée en deux blocs de 8000 Octets chacun :

```
Adresse:
 #$0000
          lignes à balayage pair: 0, 2, 4, ..., 198
              occupent 8000 Octets   ('EVEN')
 #$1F3F #7999

 #$2000 #8192
          lignes à balayage impair: 1, 3, 5, ..., 199
              occupent 8000 Octets   ('ODD')
 #$3F3F #16191

 #$4000 #16255
```

Un octet est codé pour que chaque bit représente un PEL : le premier PEL du coin supérieur gauche de l'écran est le bit 7 de l'octet 0000 (ligne 0, paire) :



SAUVEGARDE DE L'ECRAN SUR DISQUE

Un tableau d'octets ScrAr est déclaré à l'adresse correspondant à la mémoire tampon de l'affichage. C'est la procédure SaveAbsScrFile(NomF : Str14) ; où NomF est le nom du fichier écrit,
au format : 'B:????????.???'
déclaré du type : TYPE Str14 = String[14[;

RAPPEL D'UN FICHIER-IMAGE

Le rappel d'un fichier-image se fait de la même façon, par la Procedure RecScrAbsFile(NomF: Str14; Lprt: Boolean) ; Si Lprt = Vrai alors l'écran est, aussitôt après son rappel, recopié sur l'imprimante.

COMMANDES GRAPHIQUES DE L'IMPRIMANTE

L'impression graphique nécessite une modification de l'avancement interligne par la commande Esc 3 ; n pour un avancement interligne de : n × 0.1176 mm. En Turbo : Write(Lst,Chr(27), '3',Chr(24)) ;

L'avancement est remis à la normale (1/6 inch = 4.23 mm) par la commande : Esc 2 soit, en TURBO : Write(Lst,Chr(27),Chr(50)) ;

Le passage en Mode Bits d'Image Graphique est obtenu par la commande (960 octets, à vitesse normale, pour une ligne entière) : Esc Y N1 N2. Le nombre d'octets, pour une ligne, représentant des données bits d'images ('Bit Image Data'), est:
N1 + 256 × N2 (au maximum 960).
N1 est une valeur de 0 à 255.

x (7)	Chaque octet écrit après cette commande sera interprété comme un vecteur graphique vertical: chacun des 8 bits se traduit par un point s'il est à 1.
i (6)	
p (5)	Le bit de poids fort du premier octet (marqué x) doit être associé au premier PEL, bit 7 de l'octet #0000, du coin supérieur gauche de l'écran.
i (4)	
p (3)	Les bits suivants, de poids (6) à (0), de ce premier octet, correspondent aux premiers PELs de la première ligne écran. Ils se retrouvent, modulo(640=80×8), dans chacun des deux blocs EVEN (pair) et ODD (impair) de la mémoire tampon d'affichage, avec le poids 7.
i (2)	
p (1)	
i (0)	L'octet suivant destiné à l'imprimante s'obtient en prenant les bits (6) des octets d'affichage déjà considérés.

L'écran comporte 640 points PEL par ligne, N1 et N2 sont calculés : N1:= 640 Mod 256 ; N2 := 640 Div 256 ;
D'où la commande :
Write(Lst,Chr(27), 'Y',Chr(N1),Chr(N2)) ; suivie de celle d'écriture des 640 octets.

IMPRESSION D'UN ECRAN IMAGE

Cette opération est réalisée par la Procedure PrtGraph(Nol : Byte) ; NoL est le nombre de lignes-écran à imprimer (25 maximum). Les bits-écran sont pris dans le tableau d'octets absolu S (SCREEN). Les octets à imprimer sont mis dans le tableau de 640 octets P. (PRINTER). LN est un compteur de lignes.
La méthode retenue consiste à traiter 4 lignes-écran paires et 4 impaires du tableau S, pour remplir les 640 octets du tableau P. L'opération est répétée pour chaque ligne d'impression. Des pointeurs repèrent les bits, éléments de tableau et adresses.

PSE et PSO sont des pointeurs sur les octets de lignes paires (Even) et de lignes impaires (Odd) de la mémoire écran. PSB est un pointeur de bit sur ces octets. Il repère le bit à transférer dans le tableau P.

PPE est un pointeur sur P. Il repère l'octet P[PPE] dans lequel sera transféré le bit PSB de l'octet S[PSE] ou S[PSO]. PPB repère le bit de cet octet P[PPE] associé à une ligne paire ou impaire de l'écran.
PSC est un compteur d'octets (1 à 80) qui repère le traitement des octets d'une ligne-écran de S. PSB donne le numéro du bit de S[PSE ou PSO] qui doit être traité.

ALGORITHME DE LA PROCEDURE

Le processus de traitement est décrit par l'algorithme suivant :

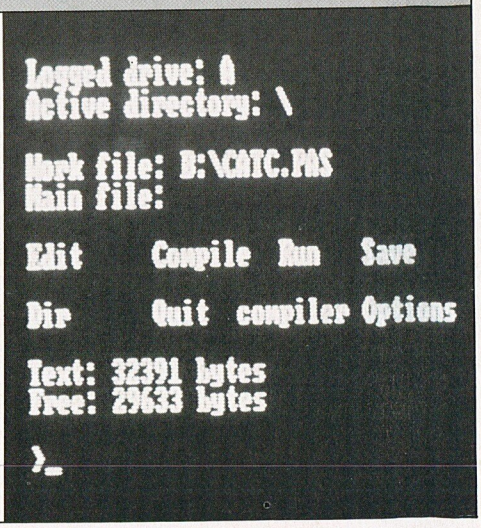
```
Initialiser LN à 1
Répéter
   {
   traitement de 4 lignes paires et 4 lignes impaires de l'écran
   }
   Initialiser PPB à 8
   Répéter
      Initialiser le tableau P à 0
      Initialiser PPE à 1
      Pour PSC de 1 à 80
         Pour PSB de 8 à 1
            Charger le bit PPB de P[PPE] avec le bit PSB de S[PSE]
            Augmenter PPE de 1
         FinDePourPSB
         Augmenter PSE de 1
      FinDePourPSC
      Diminuer PPB de 1
      Initialiser PPE à 1
      Pour PSC de 1 à 80
         Pour PSB de 8 à 1
            Charger le bit PPB de P[PPE] avec le bit PSB de S[PSO]
            Augmenter PPE de 1
         FinDePourPSB
         Augmenter PSO de 1
      FinDePourPSC
      Diminuer PPB de 1
   Jusque PPB = 0
   {
   envoi de la ligne vers l'imprimante
   }
   Commande de passage en mode graphique
   Pour PPE de 1 à 640
      Ecrire P[PPE] sur l'imprimante
   FinDePourPPE
   Augmenter LN de 1
Jusque LN > NoL
Commande de passage imprimante espacement normal
Commande de passage écran en mode texte
```

Il faut noter que, dans la procédure :

- les octets-écran S ne sont transférés vers le tableau P que si leur contenu est différent de 0.
- le transfert d'un bit-image est effectué à l'aide d'un masque M.

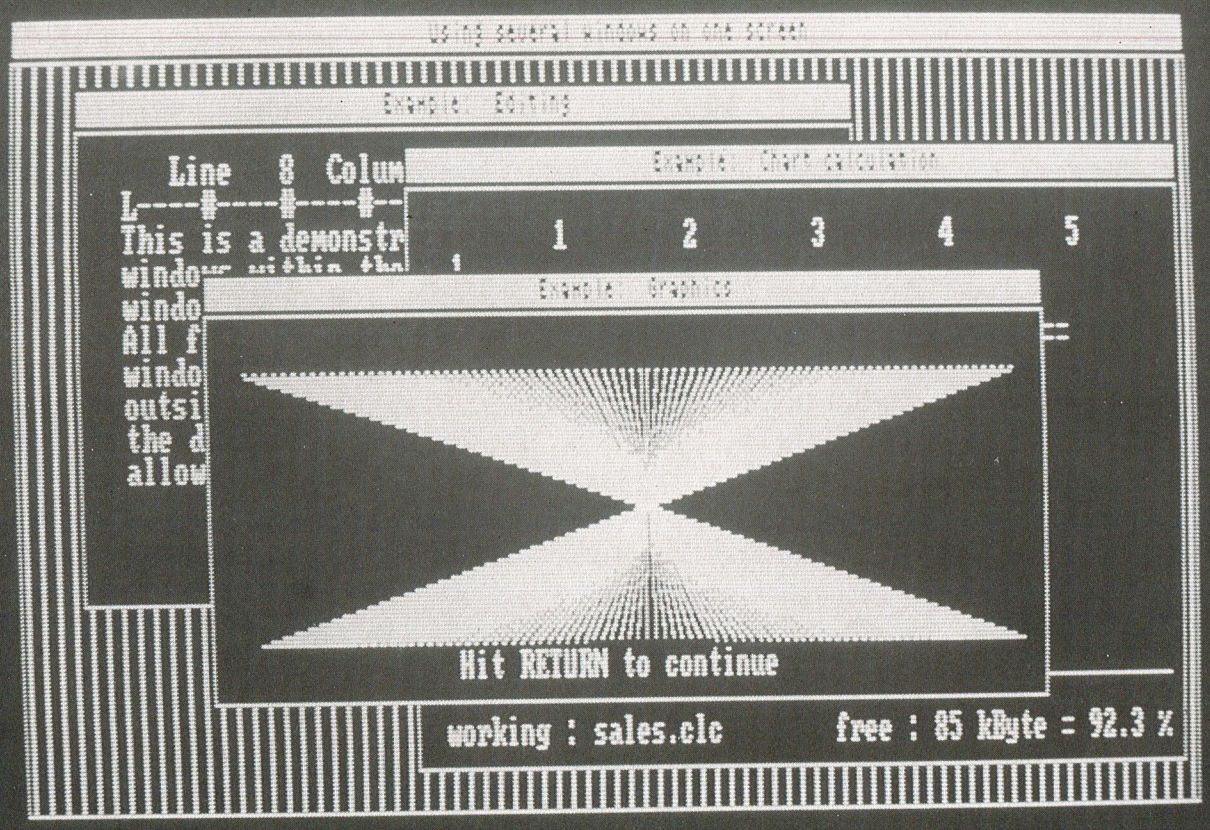
PROGRAMME D'APPLICATION

Le programme ci-joint permet de visualiser les déformations d'un carré par rotation sur une ellipse. L'angle T est augmenté d'un angle DT dont la valeur s'accroît de Pi/500 entre deux pages-écran.
La valeur de l'angle DT est affichée en bas de l'écran sur la ligne exclue de la fenêtre graphique par le '-8' de : GraphWindow(0,0,639,199-8) ; Une fois la page-écran remplie, l'utilisateur doit appuyer sur :
- 'P' s'il veut obtenir une copie sur : l'imprimante ;
- 'P' s'il veut sortir du programme ;
- n'importe qu'elle touche pour passer à l'écran suivant. Ce choix de commandes n'apparaît pas sur l'écran.



```
{   FI_LSTGR.PAS

 Fichier Inclus: Sauvegarde et Impression d'Images Graphiques

 Le mode de présentation du programme est celui de
                              la structure de l'algorithme.

}

Const
  ScrSeg = $B000;
  ScrOfs = $8000;
  EoOfs  = 16255; { fin de l'offset: 16256 - 1 }
Type
  ScrArT = Array[0..EoOfs] Of Byte; { screen array type, 127*128=16256 }
  Str14  = String[14];              { chaîne nom de fichier }

 Procedure SaveAbsScrFile( NomF : Str14);
Var
  ScrFile     : File Of ScrArT;
  ScrAr       : ScrArT absolute ScrSeg:ScrOfs;
Begin
  Assign(ScrFile,NomF); Rewrite(ScrFile);
  Write(ScrFile,ScrAr);
  Close(ScrFile);
End;

 Procedure SetScrMode(Couleur: Byte);  { 15: White }
Begin
  HiRes; HiResColor(Couleur);
End;

 Procedure RecScrAbsFile( NomF : Str14;
                          Lprt : Boolean);
Var
  ScrFile     : File Of ScrArT;
  ScrAr       : ScrArT absolute ScrSeg:ScrOfs;
  Ok          : Boolean;
  Ch          : Char;
Begin
  SetScrMode;
  Assign(ScrFile,NomF);{$I-} Reset(ScrFile); {$I+}
  Ok:=(IOResult=0);
  If Not Ok Then Begin
    TextMode; Writeln('Fichier <',NomF,'> non trouvé.');
              Writeln('............................ une touche');
    Read(Kbd,Ch); End
  Else Begin
    Read(ScrFile,ScrAr);
    Close(ScrFile);
    If Lprt Then PrtGraph(25) Else Read(Kbd,Ch);
    TextMode;
  End;
End;

 Procedure PrtGraph(NoL : Byte);  { 25 lignes maximum }
Const
  M                 : Array[1..8] of Byte
                    = (1,2,4,8,16,32,64,128); { masque }
  Nbid              = 640; { nombre de bits-image / ligne }
Type
  ScrAbAr           = Array[1..16256] of Byte;
Var
  S                 : ScrAbAr absolute ScrSeg:ScrOfs; { tableau écran }
  P                 : Array[1..Nbid] of Byte; { tableau imprimante }
  PSE, PSO, PPE,
  N1, N2            : Integer;
  LN,                 { compteur de ligne }
  PPB, PSC, PSB     : Byte;
  Res               : Boolean;
Begin
  N1:=Nbid Mod 256; { reste =  128 }
  N2:=Nbid Div 256; { quotient = 2 }
  Writeln(Lst,Chr(27),'3',Chr(24));
  PSE:=1;  PSO:=8193;  LN:=1;
  Repeat { For NL:=1 to Nol }
    PPB:=8;  FillChar(P,SizeOf(P),0);
    Repeat { Pour PPB de 8 à 1 }
      PPE:=1;
      For PSC:=1 To 80 Do Begin
        If S[PSE] <>0 Then Begin
          For PSB:=8 DownTo 1 Do Begin
            If (S[PSE] and M[PSB]) <> 0
               Then P[PPE]:=P[PPE] Xor M[PPB];
            PPE:=PPE+1;
          End;
        End Else PPE:=PPE+8;
        PSE:=PSE+1;
      End;
      PPB:=PPB-1;  PPE:=1;
      For PSC:=1 To 80 Do Begin
        If S[PSO] <> 0 Then Begin
          For PSB:=8 DownTo 1 Do Begin
            If (S[PSO] and M[PSB]) <> 0
               Then  P[PPE]:=P[PPE] Xor M[PPB];
            PPE:=PPE+1;
          End;
        End Else PPE:=PPE+8;
        PSO:=PSO+1;
      End;
      PPB:=PPB-1;
    Until PPB=0;
    Write(Lst,Chr(27),'Y',Chr(N1),Chr(N2));
    For PPE:=1 To Nbid Do  Write(Lst,Chr(P[PPE]));
    Writeln(Lst);
    LN:=LN+1;
  Until LN > Nol;
  Writeln(Lst,Chr(27),Chr(50));
End;
```

```
Program P_Carres;

Const
   Couleur : Byte = 15;

{$I B:FI_LSTGR.PAS}

Const
   Xc     = 320;  { coordonnées du centre }
   Yc     = 100;
Var
   X, Y, Xa, Ya : Integer;
   R, T, DT     : Real;
   Ca           : Char;
   Fini         : Boolean;

Begin
   Fini:=False;  DT:=Pi/2;
   Repeat
      SetScrMode(Couleur);
      GotoXY(3,25);
      Write(' POLYGONE 4 COTES SUR SPIRALE, Variation Angle: ',DT,' Radians');
      GraphWindow(0,0,639,199-8);
      Xa:=Xc; Ya:=Yc; R:=0; T:=0;
      Repeat
         X:=Trunc(Xc + 2.5*R*Sin(T));
         Y:=Trunc(Yc + R*Cos(T));
         Draw(Xa,Ya,X,Y,Couleur);
         Xa:=X; Ya:=Y; R:=R+1.5; T:=T+DT;
      Until (R>Yc) Or KeyPressed;
      Read(Kbd,Ca); Ca:=UpCase(Ca);
      If Ca='P'  Then PrtGraph(25);
      DT:=DT+Pi/500;
   Until Ca='F';
   TextMode;
End.
```



MULTIPLAN 2.0

Si vous êtes français et possesseur d'un PC et en train de lire cet article, il y a de fortes chances pour que vous ayez travaillé avec Multiplan, et même pour que vous soyez un fan de Multiplan ou même que vous ayez fait vos premiers pas sur un PC avec ce tableur. La popularité de ce logiciel dans notre pays n'est plus à démontrer et il arrive en général très largement en tête dans la plupart des enquêtes effectuées auprès des utilisateurs et pareillement demeure le logiciel le plus vendu sur le territoire français à ce jour.

Daniel ARCANES

Nous avions eu droit à une révision de la version 1.0 qui nous a amené la souris et les couleurs. Elle n'apportait pas suffisamment de mises à jour pour se positionner face à son concurrent direct, j'ai nommé 1-2-3, car il lui manquait bien évidemment les macros. Les macros dont l'absence se faisait de plus en plus ressentir dans Multiplan étant donné qu'une bonne partie des fonctions de ce

PUISSANT
RAPIDE,
SÛR...

512 Ko
le PC européen 100 % compatible
à un prix d'attaque.

MAGNUM
série 4.

une production
euro computer
B.P. 106 38.76.67.66
45240 LA FERTE SAINT-AUBIN

Création Studio Gérard DARIC

Importateur
de compatibles
PC - XT - AT
(Prix très compétitifs)
Recherche :
Distributeurs (toute
la France, Suisse,
Belgique...)
Très fortes remises
et soutien commercial
(publicité, marketing,
et politique de vente
de soft très agressive)
Ecrire :
Revue de l'Utilisateur
de l'IBM-PC
2, rue d'Amsterdam
75009 Paris
(Annonce n° A01)
qui transmettra

"Mon IBM PC AT et moi : quelle équipe musclée !"

Avec le dernier-né de la famille IBM PC, vous allez former une équipe vraiment très, très musclée.

Il a été conçu pour vous permettre de traiter une montagne d'informations, trois fois plus vite que les premiers ordinateurs personnels IBM.

L'IBM PC AT est proposé avec un disque fixe de 20 millions d'octets, ce qui vous donne assez de place pour saisir 10.000 pages d'informations ou retenir la liste de vos 200.000 meilleurs clients. Si vos besoins de stockage s'accroissent, il vous suffit d'ajouter un autre disque fixe de 20 millions d'octets.

Imprimante IBM 4201

Une telle richesse d'information demande a être protégée. C'est pourquoi l'IBM PC AT possède un verrouillage à clé.

Avec une mémoire vive de 256 kilo-octets ou plus, votre IBM PC AT vous permet d'accéder à des applications complexes, dans le domaine graphique par exemple. Et si les logiciels dont vous vous servez le permettent, vous pouvez pousser votre IBM PC AT jusqu'à 3 millions d'octets.

Conçu pour la performance, l'IBM PC AT ne rechigne pas pour autant aux travaux courants. Il accepte pratiquement tous les programmes écrits pour la famille IBM PC.

Appelez le (1) 47.22.22.22 pour avoir la liste des points de vente où vous pourrez essayer l'ordinateur personnel IBM AT et vous faire conseiller par votre distributeur agréé IBM PC.

IBM

LITTLE TRAMP CHARACTER LICENCED BY BUBBLES, INC., S.A.

RÉALISATION DUPUY-SAATCHI ÉQUIPEMENT

de feuilles de calcul créées avec 1-2-3 et je dois dire que le résultat était pour le moins bizarre. Les caractères avec accents tels que é ou ù créés sous Lotus apparaissent sous une apparence assez étrange sous Multiplan. Autre différence : les libellés qui sous Lotus 1-2-3 se continuent naturellement dans la cellule suivante s'ils dépassent la taille de la première cellule, n'apparaissent que dans la cellule d'origine sous Multiplan 2. De plus, Multiplan n'en conserve semble-t-il que 32 caractères bien qu'il soit en mesure d'enregistrer potentiellement 255 caractères par cellules.

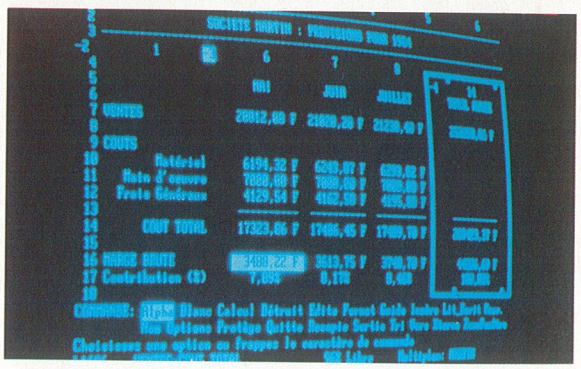
MACROS ET INSTRUCTIONS

L'inspiration 1-2-3 est visible à plusieurs endroits pour la définition des macros. La définition d'une macro peut se continuer à la ligne suivante à partir du moment où vous restez sur la même colonne. Et l'appel de la macro se fera en appuyant sur la touche Alt suivi d'une lettre ou... plusieurs ce qui permet de définir un nombre de macros assez important. Il nous sera même possible de récupérer des macros créés sur une autre feuille avec la commande eXterne. La définition d'une macro se fait par conséquent en la frappant dans une colonne. Puis nous devons appeler la commande Nom. Cette commande contient deux nouveaux paramètres lesquels sont Macro : Oui(Non) et Code d'accès. Il nous suffit d'indiquer Oui pour le premier paramètre pour indiquer que les cellules en question définissent une macro. J'ai dit les cellules et non pas la cellule de départ seule car Multiplan 2 nécessite que l'on indique toute l'étendue des cellules sur lesquelles s'étend la macro dans la commande Nom. Dans la partie Code d'accès, nous introduisons le code d'accès qui servira à appeler la macro.
Un bon point pour Multiplan 2 : le fait d'appuyer sur la touche F5 nous fait passer en mode « Edite-Macro ». Dans ce mode lorsque nous définissons une macro, il nous suffit d'appuyer sur une touche pour que l'instruction qui lui correspond s'inscrive. Un exemple 'tb est utilisé pour simuler une frappe de la touche Tabulation. Lors de la définition de la macro, si nous sommes en mode « Edite Macro », il nous suffira d'appuyer sur Tab pour que la mention 'tb s'inscrive dans notre macro. Simple et efficace. Shit et F5 nous fait passer en mode Pas à pas et nous permet d'exécuter notre macro de cette façon, ce qui permet comme dans les autres tableurs de repérer où se trouve les erreurs dans la macro. Quelques avantages qui sont eux aussi liés à la conception de Multiplan. La commande Vers permet de retrouver où se trouve les macros d'une feuille donnée (je pense qu'il a pu vous arriver de ne plus savoir où vous aviez rangé une macro dans une feuille de calcul). La commande Nom vous permet quand à elle de faire défiler les noms et donc également les noms de macros et donc de retrouver celles ci si vous ne savez plus quelles sont celles que vous aviez définies dans une feuille. Autre point commun avec 1-2-3 : il est possible de définir une macro qui sera appelée automatiquement lors du chargement d'une feuille. Ce qui est défini avec /0 sous Lotus se définit avec le nom autoexéc sous Microsoft.

Nous retrouvons aussi le caractère ? pour signifier à un utilisateur qu'il doit introduire des données.
Nous allons étudier et commenter quelques macros développées sous Multiplan 2 afin de faire ressortir les qualités et les défauts de ce tableur nouvelle version.

COMPTE DE BILAN

Les macros de Multiplan 2 comme celles d'autres tableurs peuvent être chargées en autoexecution, ce qui signifie qu'elles peuvent alors s'exécuter automatiquement lors du chargement d'un fichier donné. Dans l'exemple que nous allons voir ci dessous, c'est cette technique qui a été utilisée. Il suffit de nommer une macro AUTOEXEC lorsqu'on la crée avec la commande Nom pour en faire une macro autoexécutable. Il faudra bien sûr aussi préciser dans la commande Nom que la zone référencée est une macro et non une référence de cellule habituelle. Les macros autoexécutables présentent des avantages et des inconvénients. L'avantage est qu'elles se déroulent sans que l'on ait à intervenir à peine la feuille est chargée. Dans l'exemple que nous allons étudier, nous avons deux feuilles de calcul MAT1 et MAT2 qui se font constamment appel l'une à l'autre. Cela s'exécutera sans que nous ayons à intervenir, chacune d'elles étant dotée d'une macro AUTOEXEC. Nous verrons un peu plus loin que cela présente cependant un certain inconvénient pour la réinitialisation de ces feuilles. Pour pouvoir étudier les macros que nous vous présentons ici, nous vous conseillons de saisir les fichiers feuilles présentés ici tels quels et de saisir les macros qui y sont intégrés sur les lignes et colonnes que nous avons choisi. Vous aurez tout loisir par la suite d'enrichir ou de modifier ces macros à votre guise ou de les adapter à des situations plus spécifiques à vos besoins.
La première feuille que nous allons voir s'appelle MAT2. C'est celle qui contient les données que l'utilisateur peut être amené à changer. Dans la colonne 1, nous avons des libellés de comptes à porter au bilan, et dans la deuxième colonne nous avons le montant correspondant à chacun de ces comptes.



tableur pouvaient être rendues automatiques. Une fois passé l'effet de surprise et l'engouement qui faisait suite à la découverte des tableurs, l'utilisateur moyen se retrouvait frustré de ne pas pouvoir automatiser certaines tâches et pouvait être tenté d'aller voir ailleurs : 1-2-3, Framework, Supercalc et d'autres étaient alors des choix possibles. Multiplan possédait un environnement de travail assez riche pour donner envie d'écrire des applications basées sur ce modèle. Grâce aux macros, nous pourrons donc réaliser des modèles de tableurs avec des menus appropriés et des instructions spécifiques destinées aux utilisateurs telles que « Frappez F2 pour vous déplacer à la cellule suivante ». Il était donc urgent, en dépit de son succès national que Multiplan intègre les macros et un langage de programmation. Nous verrons que ce langage de macros emprunte d'ailleurs beaucoup à celui de 1-2-3 par certains aspects mais nous ne nous en plaindrons pas car ce sont des aspects positifs qui ont été retenus. Une seule remarque, et qui est étrange si l'on considère que Multiplan nous avait habitué à la simplicité face à son concurrent direct : le langage des macros de Multiplan me semble moins simple à utiliser que celui de 1-2-3. Multiplan a opté pour des abréviations là où Lotus utilise des mots relativement entiers et donc plus longs à coder. En définitive cela rend les macros de Multiplan plus difficiles à relire que celles de 1-2-3. Cela avec un produit qui reste tout de même sans doute le plus facile à aborder par un néophyte et que l'on retrouve toujours avec beaucoup de plaisir. L'avantage de Microsoft vient aussi de la généralisation de son type d'interface : lorsque j'ai découvert Word, je n'ai eu aucun mal à me familiariser avec ce logiciel de traitement de texte, du fait qu'il reprenait en partie dans son menu de commandes, présenté comme celui de Multiplan, des commandes similaires : Alpha, Détruit, Format, Guide, LitEcrit... Puis par la suite en reprenant Multiplan, il m'est arrivé de découvrir certaines caractéristiques reprenant ce à quoi j'avais été habitué sur Word. Exemple : la version 2 de Multiplan utilise les touches F7 et F8 pour se déplacer dans une macro de façon similaire à la fonction de ces touches sous Word : un mot en arrière ou en avant. Il en est de même avec les autres produits Microsoft tels que Chart et File : l'interface choisie par Microsoft fait que si l'on a travaillé sur l'un des logiciels de cette famille, il est assez facile de se familiariser avec les autres.

UNE DISQUETTE FERA L'AFFAIRE

Le nouveau Multiplan nous est livré avec un manuel assez proche de l'ancien : nous retrouvons entre autres dans la première partie « Comment utiliser Multiplan » les fameux établissements Martin et leur feuille de calcul. Le nouveau manuel intègre cependant les nouveaux écrans de présentation de Multiplan 2, qui sont en fait pratiquement les mêmes que l'ancien sauf pour quelques commandes telles que Nom ou Vers qui incluent les macros. Nous avons droit à une annexe supplémentaire pour nous initier aux macros. Cette annexe est cependant un peu succinte et nous tenterons dans cet article d'expliciter un peu plus les macros sous Multiplan 2.

Une seule disquette programme nous est fournie plus une copie de sauvegarde. Nous n'avons pas à procéder à une installation quelconque pour lancer le produit, ce qui se comprend du fait que Multiplan ne gère pas d'affichage ou d'impression graphiques. Nous pouvons donc l'utiliser tel quel immédiatement.

Les macros ne sont pas la seule nouveauté de ce progiciel. La taille du tableur est passée à 4.095 lignes sur 255 colonnes et la vitesse de calcul a été améliorée et cela est vraiment observable sans même effectuer de test minuté, mais il faut dire que le premier Multiplan était plutôt lent. Multiplan 2 ne gère en mémoire que les colonnes dans lesquelles se trouvent des informations, technique qui a été reprise dans Symphony, 1-2-3, et Framework. Il gère le coprocesseur 8087 ou 80287 lorsque celui ci est présent. Il propose un nouvel outil de consolidation qui permet de combiner plusieurs tableurs ayant le même format, en une seule commande. Et une autre particularité que j'ai apprécié est le fait que l'extension du pointeur de cellules lors de l'utilisation d'une commande apparaît à l'écran. Ce n'est qu'un détail, mais il manquait à la version 1 lorsque l'on désirait utiliser les touches fléchées pour marquer un champ à l'écran.

SENS UNIQUE

Microsoft a bien appris la leçon de sa coopération avec IBM : compatibilité vers le haut, la version ultérieure étant en mesure de récupérer les fichiers créés avec les versions précédentes. Nous pouvons donc lire un fichier créé avec la version 1.1 et c'est ainsi que nous avons commencé notre incursion dans la version 2.0. Première surprise : lors de la lecture du fichier une petite mention nous apparaît au bas de l'écran : « Lecture de la ligne 1 » le numéro évoluant à toute vitesse jusqu'à la mention « Lecture de la ligne 67 ». Soudain, et cela est assez étrange, cette lecture du fichier se ralentit et de même par conséquent l'affichage de la mention en question et progresse jusqu'à la lecture de la ligne 105. Ce qui est étonnant, c'est que le tableur en question ne comportait que 31 lignes. Il est vrai en revanche qu'il comportait deux fenêtres. Première chose : nous allons profiter du fait d'être sur la version 2.0 pour colorer nos fenêtres ce qui est possible avec ZoneFenêtre Mise en Couleur. Il nous est possible de choisir la couleur d'avant plan, d'arrière plan et même celle du contour si nous en mettons un à la fenêtre. Très bon point donc même s'il était déjà présent sur la dernière version de Multiplan 1. En revanche, si nous cherchons à relire avec Multiplan 1 un fichier coloré avec la version 2, Multiplan 1 s'avérera incapable de le relire. Par curiosité nous repassons sur la version 2 et supprimons les couleurs de nos fenêtres. Pas d'issue possible : la version 1 n'est pas apparemment en mesure de lire les fichiers recopiés avec la version 2. Certains diront qu'ils n'en voient pas l'intérêt à partir du moment où la nouvelle version existe.

Multiplan possédait un mode nous permettant de charger des feuilles qui avaient été créées avec VisiCalc. La version 2 va plus loin et nous annonce que dorénavant nous pouvons lire des feuilles de calcul créées avec 1-2-3 ou Symphony. Nous avons donc procédé à la lecture

Afin de déterminer si un compte est un compte d'Actif ou de Passif, nous avons dans cette même feuille de calcul MAT1 en colonne 11 et 12 un tableau des comptes et de leur appartenance. Les comptes de Passif apparaissent en tête, suivis par les comptes d'Actif. Pour que la macro puisse s'exécuter correctement, il faudra inscrire dans la feuille MAT2 uniquement des libellés présents dans ce tableau et dans leur orthographe exacte telle qu'indiquée dans ce tableau de MAT1.

```
                    11                    12
 1
 2 Les Comptes et leurs appartenance
 3
 4
 5 Capital social                P
 6 Primes d'émission             P
 7 Réserve légale                P
 8 Autres réserves               P
 9 Provisions                    P
10 Subventions d'équipement      P
11 Emprunts                      P
12 Fournisseurs                  P
13 Avances clients               P
14 Personnel                     P
15 Impots et taxes               P
16 Effets à payer                P
17 Frais d'établissement         A
18 Terrains                      A
19 Constructions                 A
20 Matériel et outillage         A
21 Matériel de transport         A
22 Mobilier et matériel de burA
23 Agencements et installationA
24 Autres immobilisations        A
25 Dépots et cautionnements      A
26 Acomptes fournisseurs         A
27 Clients                       A
28 Avances au personnel          A
29 Autres débiteurs              A
30 Chèques postaux               A
31 Caisses                       A
```

Lorsque nous allons lancer la macro à partir de MAT2, celle ci va pointer sur le premier compte présent dans cette feuille de calcul et sur le montant correspondant. Dans l'exemple ci-dessus , nous trouvons Capital et le montant 144.284.600,00 F. Ce libellé va être recopié dans la feuille MAT1. Mais en même temps, dans MAT1, une recherche va être effectuée dans le tableau des comptes afin de déterminer si Capital est un compte Actif ou Passif. Nous trouvons « P » dans le tableau, et par conséquent, le montant correspondant est recopié dans la colonne Passif.

```
-1          1          2          3          4          5          6
  1
  2 Société MAGIC SYSTEM
-2
  6                    |        ACTIF        |        PASSIF        |
  7 ------------------------------------------------------------------
  8 Capital social     |                     |     144.284.600,00 F  |
```

```
1                    1                          2                    3
  1 Libellé des
  2 Comptes à porter                                Montant
  3 au bilan
  4 ==========================                      =======
  5
  6 Capital social                          144.284.600,00 F
  7 Constructions                           103.157.067,14 F
  8 Fournisseurs                             99.280.984,87 F
  9 Clients                                 155.446.083,72 F
 10 Personnel                                   788.783,12 F
 11 Avances au personnel                        110.890,85 F
 12 Fin
 13
 14
 15
 16
 17
 18
 19
 20
OMMANDE: Alpha Blanc Calcul Détruit Edite Format Guide Insère Lit_Ecrit Mouv.
         Nom Options Protège Quitte Recopie Sortie Tri Vers Xterne ZoneFenêtr
noisissez une option ou frappez le caractère de commande
1C3                                          97% Libre EP Multiplan: b:mat2
```

La deuxième feuille de calcul MAT1 contient un tableau avec trois colonnes principales. La colonne 1 va servir à recopier les libellés des comptes extraits de la feuille MAT2. La colonne 3 servira à inscrire les montants à porter à l'Actif et la colonne 5, les montants à porter au Passif. Voici la feuille de calcul MAT1 telle qu'elle apparaît en début de travail. Nous avons d'ailleurs inclus en plus de l'AUTOEXEC une macro permettant de réinitialiser cette feuille de calcul mais nous y reviendrons.

```
1          1          2          3          4          5          6
  1
  2 Société MAGIC SYSTEM
-2
  1
  2 Société MAGIC SYSTEM
  3
  4                    Bilan au 31 juillet 1986
  5
  6                    ACTIF                    PASSIF
  7
  8
  9
 10
 11
 12
 13
 14
 15
OMMANDE: Alpha Blanc Calcul Détruit Edite Format Guide Insère Lit_Ecrit Mouv.
         Nom Options Protège Quitte Recopie Sortie Tri Vers Xterne ZoneFenêtre
noisissez une option ou frappez le caractère de commande
3C1                                          93% Libre    Multiplan: b:mat1
```

MAT2

Nous allons passer en revue les macros définies dans MAT2, la feuille qui reçoit des libellés et des valeurs de montant. Cette feuille contient trois macros : BILAN, AUTOEXEC et INIT. Pourquoi avoir défini trois macros plutôt qu'une seule macro complexe. Il y a eu plusieurs raisons à cela. La première est que les macros de MAT1 définies comme une seule macro sur plus de 100 lignes au départ n'arrivaient pas à s'exécuter. Microsoft m'a confirmé qu'un problème avait été constaté avec les macros figurant sur des feuilles trop longues et que le service technique s'employait à y remédier en ce moment même. En attendant, la solution trouvée a été de découper la macro en plusieurs macros et de les faire se débrancher les unes aux autres avec VM qui est l'initiale de Vers Macro. Par la suite, j'ai trouvé que cela ajoutait beaucoup de clarté à la présentation des macros.

Les macros présentées ici sont toutes sous la forme suivante : le titre est indiqué sur la colonne de gauche de la macro en majuscules à titre de référence uniquement. A gauche de la macro se trouve un commentaire. Chaque macro est inscrite dans une seule et même colonne sans jamais déborder celle ci. De plus, presque chaque ligne de macro se termine par 'rt ce qui correspond à la frappe de la touche ENTREE au clavier. J'ai trouvé qu'il s'agissait là d'un découpage en unités d'actions simples assez logique. Observons la macro **AUTOEXEC** de MAT2 :

```
15 AUTOEXEC   VL118'tbC3'rt                  Nom Auto en L122 C3
16            NAuto'tbL122C3'rt              Vers Nom Auto
17            VNAuto'rt                      Si Auto=OUI => Vers Macro Bilan
18            'siAuto=L118C3'VMBilan'rt      Sinon demande entrée réponse
19            A'?'rt
20            'siAuto=L118C3'VMInit'rt       sinon fin macro
21            'or                            home
22            'qu                            fin
23
```

La première ligne de la macro AUTOEXEC correspond à la frappe des touches Multiplan Vers Ligne 118 Colonne 3. La ligne suivante donne le Nom Auto à la cellule L122C3. Observons ce que nous avons dans cette partie de l'écran :

```
1              3              4                    5
116
117
118          OUI
119            /
120          NON
121           v                    Désirez vous lancer l'
122          OUI                   exécution automatique
123
124
125
126
127
128
129
130
131
132
133
134
135
_PHA: OUI

ntrez votre texte (sans guillemets)
122C3     "OUI"                          97% Libre     Multiplan: b:mat2
```

De même, le deuxième libellé de MAT2, « Constructions » va être recopié dans MAT1. Après recherche dans le tableau, le montant correspondant sera porté dans la colonne Actif.

```
-1          1          2          3          4          5          6
   1 Total
   2 Société MAGIC SYSTEM
-2
   3
   4                    Bilan au 31 juillet 1986
   5
   6                    ACTIF                    PASSIF
   7
   8 Capital social                              144.284.600,00 F
   9 Constructions        103.157.067,14 F
  10 Fournisseurs                                99.280.984,87 F
  11 Clients              155.446.083,72 F
  12 Personnel                                      788.783,12 F
  13 Avances au personnel     110.890,85 F
  14
  15 Total                258.714.041,71 F       244.354.367,99 F
  16 Bénéfices                                    14.359.673,72 F
  17
COMMANDE: Alpha Blanc Calcul Détruit Edite Format Guide Insère Lit_Ecrit Mouv.
          Nom Options Protège Quitte Recopie Sortie Tri Vers Xterne ZoneFenêtre
Choisissez une option ou frappez le caractère de commande
L17C1                                  93% Libre    Multiplan: mat1
```

Le même processus va se répéter entre les deux feuilles de calcul jusqu'à ce que le libellé « Fin » soit repéré dans la feuille MAT2. A partir de ce moment là, les totaux vont être calculés et la macro de MAT1 va déterminer si le bilan est positif ou négatif et inscrire le montant des Pertes ou des Bénéfices. Dans le cas présent, nous pouvons observer que la macro s'est terminé par un solde bénéficiaire, mais vous pourrez la tester avec un solde déficitaire afin d'observer la sortie à l'écran dans ce cas-là.

```
-1          1          2          3          4          5          6
   1
   2 Société MAGIC SYSTEM
-2
   6                    ACTIF                    PASSIF
   7
   8 Capital social                              144.284.600,00 F
   9 Constructions        103.157.067,14 F
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
COMMANDE: Alpha Blanc Calcul Détruit Edite Format Guide Insère Lit_Ecrit Mouv.
          Nom Options Protège Quitte Recopie Sortie Tri Vers Xterne ZoneFenêtre
Choisissez une option ou frappez le caractère de commande
L9C5                                  93% Libre    Multiplan: mat1
```

à permettre une relecture plus facile de la macro. Dans la ligne de commentaire, nous retrouvons la mention de ces étiquettes là où cela s'applique. Nous trouvons par exemple aux lignes 25 et 26 de la colonne 19 : « ... branche à PASSIF ». Cela est quand même plus parlant que l'instruction de débranchement : « peL37C18 ». Ces étiquettes sont artificielles étant donné qu'elles ne sont pas traitées par Multiplan mais elles relèvent d'habitudes de programmations qui veulent qu'il faut chercher à rendre les programmes aisés à relire. Je dois dire au passage que j'ai été assez gêné par le fait que Multiplan 2 ne permet pas le branchement à un nom de cellule mais seulement à une adresse de cellule. Chaque fois que j'ai donc déplacé une partie de texte de macro, il m'a fallu réajuster toutes les adresses des instructions 'pe, lesquelles servent au débranchement.

La ligne 18 contient les textes des macros. Les lignes 'co sont très nombreuses ici, souvent uniquement utilisées pour mettre des titres de paragraphes. La ligne 19 achève de commenter cette macro. MAT1 contient 5 macros : AUTOEXEC, SUITE, RETOUR, ACTPAS, et TOTAL.

AUTOEXEC est une macro qui fait pointer les cellules nommées Compt, Actif, et Passif une ligne plus bas à chaque appel sur la feuille MAT1. A la ligne 11, nous avons un Xterne Recopie de la feuille MAT2. Le libellé désigné par Libel sous MAT2 est recopié dans la cellule Compt de MAT1. Si Compt contient « Fin », c'est la fin de la récupération des données en provenance de MAT2, et l'on se débranche à TOTAL. Dans le cas, contraire, on passe à SUITE.

SUITE a pour seul but de nous débrancher vers la macro ACTPAS. Elle a été introduite ici afin de rendre plus claire une lecture linéaire du listing.

ACTPAS est une macro qui prend le libellé récupéré dans Compt et effectue une recherche dans le tableau des comptes et de leur appartenance au Bilan. ACTPAS teste chaque ligne du tableau et la compare à Compt. Si elle ne trouve pas correspondance, le sous programme Dessous va nous permettre de descendre d'une ligne. Lorsque le compte est trouvé dans le tableau, ACTPAS passe la main à RETOUR.

RETOUR en fonction de la ligne du tableau où la correspondance a été trouvée inscrit le code A ou P dans une cellule nommée Pasact. Selon la valeur de Pasact, nous allons alors recopier le montant présent dans MAT2 dans la colonne ACTIF ou PASSIF de MAT1. Une fois ceci effectué, nous rechargeons MAT2 et le traitement repart à la ligne suivante de MAT2.

TOTAL effectue une totalisation des colonnes Actif et Passif et en fonction du rapport de forces Actif / Passif, inscrit la différence des deux dans la colonne correspondante en indiquant le libellé « Pertes » ou « Bénéfices ».

```
     16        17                    18                          19
 1 Variable Nom macro             Texte                Commentaires
 2
 3          AUTOEXEC VNCompt'rt                        Vers Nom Compt (col. 1)
 4                   NCompt'tb'vb'rt                   Donne le nom Compt à
 5                   'co                               la cellule du dessous
 6                   VNActif'rt                        Vers Nom Actif (col. 3)
 7                   NActif'tb'vb'rt
 8                   VNPassif'rt                       Vers Nom Passif (col. 5)
 9                   NPassif'tb'vb'rt
10                   'co ------------------------------
11                   XRmat2                            Xterne Recopie mat2
12                     'tbLibel'tbCompt                  Libel (mat2) dans Compt
13                     'tbN'rt                           Non couplé
14                   'co ------------------------------
15 Fin                'siCompt<>L15C16'VMSuite'rt si Compt='Fin' branchement
16                   VMTotal'rt                        Vers Macro TOTAL
17
18          SUITE    VMActpas'rt                       Vers Macro ACTPAS :
19                   'co                               détermine Actif ou Passif
20                   'co                               dans le tableau cols 11-12
21
22          RETOUR   VNTest'rt                         Test = compte référencé
23                   'co                               dans le tableau col 11
24                   NPasact'tb'vd'rt                  Pasact = code A/P (col 12)
25 P                 'siPasact=L25C16''peL37C18'       si Pasact='P' branche
26                   'co                               à PASSIF
27                   'co ------------------------------
28                   'co Trait Compte ACTIF
29                   'co ------------------------------
```

Un texte nous demande si nous désirons lancer l'exécution automatique. Nous pourrons répondre par OUI ou par NON. Si nous ne répondons pas OUI, la macro s'arrêtera là. Avantage de cette programmation : nous pourrons appeler MAT2 et entrer de nouvelles valeurs avant de relancer une exécution automatique. Nous voyons que deux tests SI s'effectuent dans la macro AUTOEXEC. Le contenu de la cellule L122C3 que nous avons appelé Auto est comparé à celui de la cellule L118C3 laquelle contient OUI. Le premier test nous dit donc que si Auto contient OUI, nous allons nous débrancher Vers la Macro Bilan. Sinon, nous passons à la ligne du dessous. Le A correspond à la commande Alpha et le point d'interrogation indique qu'une réponse est attendue de l'utilisateur.

Si celui ci répond Oui, nous sommes débranchés vers une macro d'initialisation des paramètres de la feuille. Si l'utilisateur répond NON, la macro se termine, le 'qu étant l'instruction permettant de QUitter une macro.
Observons le contenu de la macro **INIT** :

```
24 INIT      N                          Nom Libel en L5C1
25            Libel'tbL5C1'rt
26           N                          Nom Montan en L5C2
27            Montan'tbL5C2'rt
28           VMBilan'rt                 Vers Macro Bilan
```

Cette macro utilise la commande Nom de Multiplan. Elle positionne deux zones Libel et Montan à la ligne 5 de la feuille MAT2. Cela nous permet de démarrer juste avant la première ligne des libellés. La macro INIT se débranche ensuite sur la macro BILAN. Si l'on casse la macro par le fait de presser Esc avant qu'elle se soit terminée, il faudra relancer INIT individuellement pour faire repartir l'exécution au début de la feuille, ou bien encore remettre à blanc la cellule nommée Auto vue plus haut.

```
        4                    5                    6
 1            Macro de MAT2                 commentaires
 2            ==============                ============
 3 BILAN      VNLibel'rt                    Vers Nom Libel
 4            NLibel'tb'vb'rt               Nom Libel TAB BAS :
 5            'co                           Donne le nom Libel à la
 6            'co                           cellule du dessous
 7            'SiLibel=L7C3'BL122C3'rt      Si Fin raz cellule initialisat.
 8            'co ---------------------------
 9            VNMontan'rt                   Vers Nom Montant
10            NMontan'tb'vb'rt              Nom Montan TAB cellule dessous
11            'co ---------------------------
12            LCmat1'rt                     Lit_Ecrit Charge mat1
13            OO'rt                         Sauv - écras : Oui
14
```

La macro **BILAN** de MAT2.
Dans cette macro, tout comme dans les macros de la feuille MAT1, j'ai abusé de l'instruction 'co qui sert à inscire un commentaire dans une macro. Ce qui peut sembler inutile à première vue, a été fait intentionnellement afin de permettre de lire aisément ces macros et de pouvoir indiquer en regard de la colonne macro des commentaires couvrant éventuellement plusieurs lignes. Essentiellement, cette macro va consister à chaque fois qu'elle est appelée à faire pointer Libel et Montan une ligne plus bas. Lors du premier appel, Libel va pointer sur Capital, puis lors de l'appel suivant sur Constructions... La ligne 7 de BILAN comporte un test comparant Libel avec le contenu de la cellule L7C3. Cette cellule est une cellule de référence et contient la valeur « Fin ».

Cela n'apparaît pas sur les écrans reproduits ici, il vous faut donc l'introduire pour que la macro effectue intégralement son travail. Le but de ce test effectué à la ligne 7 de BILAN consiste à remettre à blanc la cellule nommée Auto vue plus haut une fois que le libellé « Fin » a été détecté.

Ainsi lors du prochain appel de la macro MAT2, nous pourrons à nouveau choisir l'exécution automatique ou non. A la fin de BILAN, nous avons la lecture de MAT1. Cette feuille de calcul va se charger et s'exécuter automatiquement.

MAT1

Nous avons réparti la description des macros de cette feuille sur 4 colonnes. La première contient les variables éventuellement utilisées par la macro en référence. La cellule 125C16 qui contient la valeur P en est un exemple.
Sur la colonne 17, nous avons les noms des macros là aussi à titre de référence. AUTOEXEC est indiqué en L3C17 mais bien sûr il faudrait indiquer L3C18 à Multiplan comme point de départ. Plus loin dans la colonne 17, nous trouvons des mentions telles que Passif à la ligne 37 ou Boucle à la ligne 57. Ce sont des étiquettes artificielles destinées

POUR MIEUX CONNAITRE VOTRE IBM-PC
OU VOTRE COMPATIBLE

Profitez de notre offre spéciale d'abonnement à

OFFRE SPECIALE

Bulletin d'abonnement

à adresser à :
SOFTIN COMMUNICATION
2, rue d'Amsterdam
75009 Paris

M. .....

.....

Adresse .....

.....

souhaite bénéficier de l'offre spéciale d'abonnement à IBM-PC et COMPATIBLES HEBDO au prix spécial de 350 F TTC au lieu de 395 F TTC

☐ chèque ci-joint

☐ à facturer

```
30                  XRmat2                          Xterne Recopie mat2
31                    'tbMontan'tbActif               Montan (mat2) dans Actif
32                    'tbN'rt                         non-couplé
33                  LS'rt                           Lit_Ecrit Sauvegarde mat1
34                  O'rt
35                  'an
36                  LCmat2'rt                       retour à mat2
37        Passif    'co ----------------------
38                  'co Trait Compte PASSIF
39                  'co ----------------------
40                  XRmat2                          Xterne Recopie mat2
41                    'tbMontan'tbPassif              Montan (mat2) dans Passif
42                    'tbN'rt                         non-couplé
43                  LS'rt                           Lit_Ecrit Sauvegarde mat1
44                  O'rt
45                  'an                             Esc
46                  LCmat2'rt                       retour à mat2
47
48        ACTPAS    'co ----------------------
49                  'co Détermine si un compte
50                  'co est Actif ou Passif
51                  'co ----------------------
52                  VL5'tb11'rt                     Vers Ligne 5 Colonne 11
53                  'co
54                  NTest'rt                        Nom Test
55      1           VNResult'rt0'rt                 0 dans Result
56                  'co                             BOUCLE de recherche
57        Boucle    'co ----------------------
58                  'co    Boucle
59                  'co ----------------------
60                  'siTest<>Compt''spL68C18'       s-prog DESSOUS
61                  'co                             passe cellule du

    16        17                  18                          19
62                  'siTest<>Compt''peL58C18'       dessous et boucle
63                  'co
64                  VNResult'rt1'rt                 Trouvé : 1 dans Result
65                  'an                             Esc
66                  VMRetour'rt                     fin Macro ACTPAS
67                  'co                             Vers Macro RETOUR
68        Dessous   'co ----------------------
69                  'co S-Prog : Passe à la
70                  'co cellule du dessous
71                  'co ----------------------
72                  VNTest'rt                       Vers Nom Test
73                  NTest'tb'vb'rt
74                  're
75
76        TOTAL     'co ----------------------
77                  'co   Total Actif et
78                  'co   Total Passif
79                  'co ----------------------
80                  VNCompt'rt                      Vers Nom Compt
81                  B'rt                            Blanc
82                  'vb
83                  NCompt'tb'vb'rt
84                  ATotal'rt                       inscrit 'Total'
85                  'vd'vd                          Passe en col 3 (Actif)
86                  NActif'rt
87                  CSomme(L8C3:L(-1)C)'rt          Calcul Somme colonne Actif
88                  'co
89                  'vd'vd                          Passe en col 5 (Passif)
90                  NPassif'rt
91                  CSomme(L8C5:L(-1)C)'rt          Calcul Somme colonne Passif
92                  'co
93                  'siActif<Passif''peL106C18'     branche à PERTES
94        Bénéfice  'co ----------------------
95                  'co    ACTIF >= PASSIF
96                  'co ----------------------
```

...LETTRES à ELISE...

Une gamme très professionnelle
TRAITEMENT de TEXTES
et d'autres applications bien sûr !

Système complet à 9990 F HT

+ MS DOS : 850F HT

• MICRO ORDINATEUR COMPATIBLE IBM-PC-XT*
+ IMPRIMANTE + LOGICIEL TRAITEMENT DE TEXTES

SPECIFICATIONS :

En version de base :

• **Micro ordinateur TITAN -** série B - 256 K RAM - 2 Drives 360 K - Ecran haute définition - Clavier - Interfaces - Câbles.

• **Imprimante BROTHER -** 80 col. - 50 cps et qualité courrier.

• **Logiciel ELISE -** Mailing + fusion - Recherche - Remplacement - Glossaire - Espacement proportionnel - Imp. simultané - tableaux graphiques - symboles mathématiques - 15 polices de caractère - Insertion de fichiers - Paramétrage imprimantes - Gestion souris et couleurs - Liaisons avec Dbase - Multiplan etc. Disque virtuel.

• **Options -** Ecran couleur - Disque dur 20 Mo - Imprimantes Marguerite avec introducteur feuille à feuille...

POINTS DE VENTE :

LA MAISON DU COMPATIBLE
8 bd Magenta / 42.08.12.90
75010 Paris

POLYPHOT
17, rue de la Plaine / 43.73.81.28
75020 Paris

DELTA SYS
71, rue Ste Anne / 42.86.84.96
75002 Paris

CMS
22, rue Emile Baudot / 69.30.13.79
91120 Palaiseau

EUROPE BUREAU
71, cours de France - RN7 /
69.21.40.24
91260 Juvisy

IT-KIT
536 LP Québec - ZAC / 69.28.83.50
91946 Les Ulis Cedex

HELP AUTOMATION
21, rue J. Jaurès / 46.55.42.43
92120 Montrouge

GENIOUS SYSTEME
86, rue André Morizet / 46.05.35.80
92100 Boulogne

BM-ALPHA
10 Lot des « Lamaris » / 67.47.53.77
34430 Montpellier

ESPACE ANTIBE INFORMAT.
1 Chemin St Claude / 93.65.84.37
06600 Antibes

ELECTRONIC CHARENTAISE
13, rue A. France / 45.69.35.48
16340 L'Isle d'Espagnac

ORDIN'OCCAS
64, cours de la Liberté / 78.95.48.98
69003 Lyon

C.R.I
15, av. Mendès-France / 32.51.95.56
27200 Vernon

INFORGA
57, rue Lamendin / 21.43.94.06
62880 Vandin Le Viel

Autres régions : contactez SSIMME 69.21.84.85

* Marque déposée IBM.

```
97                  VNCompt'rt                    Vers Nom Compt
98                  'co
99                  ABénéfices'rt                 inscrit 'Bénéfices'
100                 VNPassif'rt                   Vers Nom Passif
101                 'vb
102                 CActif-Passif'rt              Calcul : Actif - Passif
103                 'vg'vg
104                 B'rt                          Blanc dans Actif
105                 'peL118C18'                   branche à FIN
106         Pertes  'co ------------------------
107                 'co     ACTIF < PASSIF
108                 'co ------------------------
109                 VNCompt'rt
110                 APertes'rt                    inscrit 'Pertes'
111                 VNActif'rt                    Vers Nom Actif
112                 'vb
113                 CPassif-Actif'rt              Calcul Passif - Actif
114                 'vd'vd
115                 B'rt                          Blanc dans Passif
116                 'peL118C18'
117                 'co
118         Fin     'or                           Home
119                 ATotal'rt                     inscrit 'Total'
120                 'qu                           Quitte la macro
121
122
```

INITIATION DE MAT1

Vous pouvez tester le déroulement de ces macros, à condition de démarrer par MAT2. Les touches Ctrl et T permettent de passer en mode Pas à Pas afin de tester étape par étape des points litigieux d'une macro. Un bon point pour Multiplan 2 : nous pourrons quitter le mode Pas à Pas et y repasser à volonté pendant le déroulement d'une macro. Il existe un dernier problème lorsque l'on a exécuté toute cette séquence de macros intégralement : la remise à blanc de la feuille MAT1. Pour cela, j'ai écrit une autre macro appelée **INIT** qui fera le nettoyage des colonnes 1, 3 et 5 jusqu'à la ligne 20. J'ai enregistré cette macro avec le code d'accès i.

```
        16        17                18                        19
123
124
125
126             INIT    NCompt'tbL7C1'rt              Compte : L7 C1
127                     VNCompt'rt
128                     B'vb:L20C1'rt                 RAZ Colonne
129                     'co
130                     NActif'tbL7C3'rt              Actif : L7 C3
131                     VNActif'rt
132                     B'vb:L20C3'rt                 RAZ Colonne Actif
133                     'co
134                     NPassif'tbL7C5'rt             Passif : L7 C5
135                     VNPassif'rt
136                     B'vb:L20C5'rt                 RAZ Colonne Passif
```

Voici donc la séquence à suivre pour redémarrer nos macros après un déroulement complet :

1) Appeler MAT1 et casser le déroulement de la macro AUTOEXEC en appuyant sur la touche Esc.

2) Appeler INIT avec alt i ou avec Vers Macro Init.

3) Sauvegarder MAT1

4) Charger MAT2. Et c'est parti... Ces macros sont assez longues à se dérouler, donc soyez patients. Si elles se plantent un peu trop tôt, ce qui serait dû à une erreur de recopie, vous pouvez indiquer Fin comme deuxième libellé en MAT2 afin de cerner sur un petit exemple où se trouvent la défaillance.

A vos macros. Partez !

Note : Ces macros ont été réalisées en collaboration avec Charles Similia de Microsoft.

LE GENERATEUR D'APPLICATIONS

SUBLIME®

MONOPOSTE-MULTIPOSTES-RESEAUX

LA PUISSANCE LA CLARTE

SUBLIME, un système de développement d'une **supériorité prouvée.**

SUBLIME, générateur d'applications multi-postes et réseaux, est la seule réponse aux problèmes créés par les méthodes de développement traditionnelles et vous assure une **transportabilité** totale de vos applications.

SUBLIME, c'est de nouvelles perspectives offertes aux utilisateurs désireux de réaliser leur application de la **plus simple à la plus sophistiquée.**

COGICIEL

27 bis, Quai Anatole-France - 75007 PARIS
Tél. 45.55.18.46

O'HANLON DATA BASE SOLUTION® - O'HANLON COMPUTER SYSTEMS

Je désire recevoir une documentation sur **SUBLIME**

Nom : Prof. :

Société : Adresse :

Tél. :

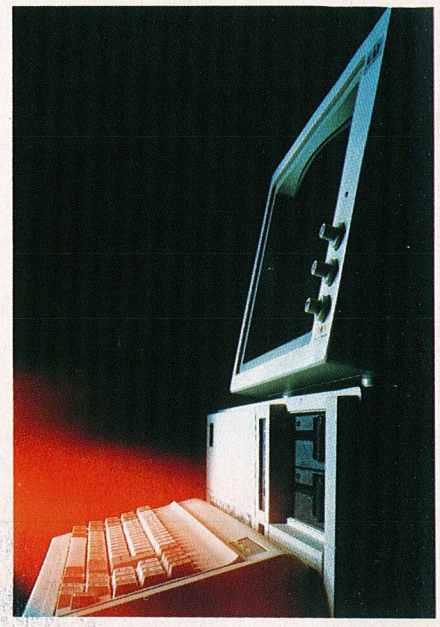
Ordinateur utilisé :

S.M. 21

Retournez ce bon à :
COGICIEL, 27 bis, quai Anatole-France, 75007 PARIS
Tél. : 45.55.18.46 - 45.55.21.66

Pour plus d'informations, cerclez 131
Service lecteur télématique p. 117

CAF : MISSION ACCOMPLIE
SUCCÈS CAF : SÉRIE PC. MASTER
L'ÉVOLUTION VERS L'UNIVERS PC/AT



PC. MASTER. 401 CONFIGURATION DE BASE

- Micro-processeur : 80286 - 6 MHZ - Timer 10 MHZ - 8 slots .
- Mémoire : 512 K extensible directement à 1024 K RAM sur carte mère et à 16000 K par adjonction de cartes.
- Contrôleur deux disquettes et disque dur.
- Une unité de disquette 5" 1/4 1200 K.
- Carte graphique couleur (sorties monochrome et couleur).
- Deux interfaces série RS. 232 et un interface parallèle imprimante.
- Clavier français AZERTY.
- Alimentation 195 W.
- DOS* 3.1.
- OPTIONS DOS : UNIX* (Venix Système 5) XENIX*. MOS*. PROLOGUE*.

PC. MASTER.402 Même configuration que PC. MASTER. 401 excepté deux unités de disquettes au lieu d'une .

PC. MASTER UN COMPATIBLE IBM/AT QUI CUMULE LES PLUS.

PC. MASTER LA MÉMOIRE D'UN MINI : 512 K à 16000 K RAM.

PC. MASTER UN CHOIX DE DISQUES : 20 Mo à 80 Mo.

PC. MASTER LA SÉCURITÉ EN PLUS GRÂCE AU STREAMER INTÉGRÉ AVEC DES SYSTÈMES D'EXPLOITATION mono et multitaches DOS 3.1 - UNIX, XENIX Prologue - MOS. ET LE SERVICE OMNIUM-PROMOTION : 12 ANNÉES D'EXPÉRIENCE INFORMATIQUE.

PC MASTER : DES PRIX « CHOCS ». PC MASTER toujours mieux et tellement moins cher

*Marques déposées - DOS : Microsoft - UNIX : Bell Laboratories - XENIX : Microsoft MOS : Interlogiciel - Prologue - IBM : International Business Machines Corporation.

PC. MASTER 403. 512 K - 1 disquette 1200 K - 1 disque 20 Mo
Même configuration que PC. MASTER 401 avec en plus un disque dur 20 Mo .

PC. MASTER 404. 512 K - 2 disquettes 1200 K - 1 disque 20 Mo
Même configuration que PC. MASTER 402 avec en plus un disque dur de 20 Mo .

PC. MASTER 443. 512 K - 1 disquette 1200 K - 1 disque 40 Mo.
Même configuration que PC. MASTER 401 avec en plus un disque dur de 40 Mo .

PC. MASTER 483 : 1024 K - 1 disquette 1200 K - 1 disque dur 80 Mo.

OPTIONS :
Streamer intégré de 20 Mo - 60 Mo.
Carte EGA : Haute résolution monochrome et couleur (640 x 350) - 7 modes de résolution écran graphique.

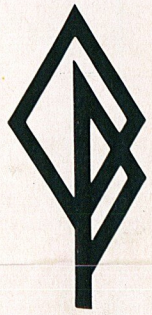
RECHERCHONS REVENDEURS

Bulletin-réponse :
Société : _____
_____
Adresse : _____
_____
Nom du responsable : _____
Tél. : _____

IMPORTATEUR EXCLUSIF EN FRANCE
OMNIUM PROMOTION

110, avenue Marceau, 92400 COURBEVOIE
Tél. 47.88.51.42 - Télex PROMIUM 610.136 F

Services Techniques : 11, rue de Cambrai, 75019 PARIS
Tél. : 42.02.17.73 - Télex PROMIUM 213.084 F



SINTELEX 103D

le système complet qui apporte la communication télex à votre IBM-PC ou compatibles

Le nouveau standard TELEX pour IBM-PC et compatibles.

CARACTÉRISTIQUES

- Saisie des messages télex sur votre traitement de textes *habituel* (Textor, Wordstar, etc.).
- Messages télex : fichiers MS-DOS. Intégration des messages dans vos bases de données...
- Tri des messages télex. Exemple : retrouver tous les messages télex envoyés à un destinataire donné ou dans une période donnée.
- Suivi de toutes les transactions grâce au Journal de Service. (Récapitulatif des envois et réceptions).
- Mémorisation ou impression des messages dans le Sintelex si l'ordinateur est occupé ou éteint.
- Fonctionnement SILENCIEUX.

OPTIONS

- Mode conversationnel.
- Envoi différé.
- Envoi d'un groupe de messages.
- Intégration dans un réseau local.
- Connection à des systèmes serveurs Minitel.

PROPRIÉTÉS TECHNIQUES

1. Se relie au PC par ligne. Série RS 232 C.
2. Transmission PC-Sintelex à 2.400 bauds.
3. Transmission sur ligne Télex 50 bauds.
4. Dimensions : 8 cm × 39 cm × 27 cm. Poids : 2 kg.

EVROZ électronique

40, rue d'Hautpoul
75019 PARIS
Tél. : 203.30.43 - Télex 216517

Importateur exclusif
Recherchons revendeurs

AGRÉÉ PTT N° 85018 T du 20.3.85. Catégorie 1.

MAINTENANCE ASSURÉE.